

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

Variability-aware design of low power SRAM memories

Promotor:
Prof. Dr. ir. W. Dehaene
Copromotor:
Prof. Dr. ir. F. Catthoor

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Stefan Cosemans

May 2009



KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

Variability-aware design of low power SRAM memories

Jury:
Prof. Dr. ir. G. De Roeck, voorzitter
Prof. Dr. ir. A. Haegemans, voorzitter
Prof. Dr. ir. W. Dehaene, promotor
Prof. Dr. ir. F. Catthoor, copromotor
Prof. Dr. ir. G. Gielen
Prof. Dr. ir. K. Maex
Prof. Dr. ir. T. Noll
Prof. Dr. ir. I. Verbauwhede

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Stefan Cosemans

U.D.C.: 621.3.049.77

May 2009

© Katholieke Universiteit Leuven — Faculteit Ingenieurswetenschappen
Kasteelpark Arenberg 1, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden, door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaandelijke schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form, by print, photoprint, microfilm or any other means without prior written permission from the publisher.

ISBN 978-94-6018-066-8

U.D.C. 621.3.049.77

Wett. Dep. D/2009/7515/49

Woord vooraf

"zonder maat"

— Frank Vander Linden

Er zijn er die beweren dat doctoreren soms eenzaam lijkt. Dat is dan alles wat er staat, één enkel zinnetje, en dan verder, relativeren, lachen en mooi zijn. Natuurlijk is het waar dat er de voorbije jaren veel dingen waren om blij, of toch om dankbaar voor te zijn. Toch zou ik de waarheid geweld aandoen door de minder geslaagde aspecten van het doctoreren achter dat ene zinnetje te verschuilen. Anders en beter, daar moeten we voor gaan, maar daarvoor moeten we ons herinneren wat er mis ging en leren uit de gemaakte fouten.

Herinneren. De maanden die wel eeuwen leken aan een terminal die hooguit geschikt bleek voor het lezen van mails en ook dan nog vaak de geest liet. Verloren tijd, frustratie. Verdwalen in literatuur die meestal niet verder geraakt dan een opbod van de grootste, de kleinste, de snelste. Geen overzicht, geen aanknopingspunt, geen begin, iedereen doet maar wat. Het eerste ontwerp dan, en de eerste layout, op een HP unix machine met een CRT waarvan de kleuren het begaven. Dag in, nacht uit, zeven op zeven, geen flauw idee waar het zou eindigen. 's Nachts rechthoekjes zien, stijf staan van de stress. En dan vlak na de tape-out nog even een aanvraag voor het IWT schrijven, met trillende handen en een inzinking nabij. Of was dat punt toen misschien al een tijdje voorbij?

Herinneren ook: het tweede ontwerp, gestart met een half goed idee en voor de rest vooral veel goede intenties. Op twee paarden wedden en er anderhalf verliezen. Sorry Anselme. Herkenbare taferelen, dag en nacht, zeven op zeven. Pijnlijk onaangepaste software, pijnlijk slechte stoel, pijnlijke rug ook. Dan de calvarietocht langs conferenties. Een njet voor ISSCC valt nog met de mantel der liefde te bedekken, maar het antwoord op mijn vraag naar het oordeel en de bemerkingen van de reviewers voor de inzending voor IEEE VLSI Circuits Symposium gaat mij een stap te ver: "Unfortunately, VLSI has a very strict policy on not giving out reviewers comments". A 1997 Ashbury Faith song comes to mind. Die strikte policy geldt ook voor andere illustere conferenties die door Widerkehr and Associates worden georganiseerd, zoals IEEE IEDM. Er is blijkbaar nog wat werk aan de winkel voor we echt mogen leren uit onze fouten. In this context, I would like to thank all JSSC reviewers who invest their time and effort to provide constructive comments.

De voorbije jaren waren soms ook mooi, en in vele opzichten erg leerrijk. Deze periode werd mogelijk gemaakt en kreeg kleur, geur en smaak dankzij de vele mensen om me heen. Ik wil hen bij deze gelegenheid dan ook graag een woordje van dank toeschrijven, soms ook wel meer dan één.

De traditie wil dat eerst de promotor wordt bedankt. Voor één keer heb ik absoluut geen bezwaar met die rangorde. Wim, bedankt voor je aanwezigheid en betrokkenheid tijdens de voorbije jaren. Als geen ander wist je binnen de lijnen te kleuren van het immer smalle, kronkelende pad tussen de verstikkende bemoedering en de verlammende veronderstelling van zelfredzaamheid. Je nuchtere kijk op de dingen was vaak onmisbaar. Ook Francky verdient een bijzonder woord van dank. Teksten en papers, hoe dicht bij de deadline ik ze ook doorstuurde, werden altijd tijdig voorzien van diepgravende verbeteringen en suggesties. Francky's niet aflatende holistische pogingen om circuit-niveau en systeem-niveau ontwerp op elkaar af te stellen hebben zeker bijgedragen tot de kwaliteit en de toepasbaarheid van dit werk. Daarnaast zou ik ook de overige leden van de begeleidingscommissie en van de jury willen danken: Georges Gielen, Karen Maex, Tobias Noll en Ingrid Verbauwhede. Dank voor het doorworstelen van het manuscript en voor de constructieve vragen en bemerkingen. Special thanks to Tobias Noll for the encouraging words. Ook dank aan de voorzitters professor De Roeck en professor Haegemans.

Zonder de nabije aanwezigheid van IMEC zou het veel moeilijker zijn geweest om het verschroeiende tempo van deze industrie bij te benen. Dank aan iedereen die bij SLI en TAD betrokken was, in het bijzonder Antonis Papanikolaou, Bamal Mandeep, Bertrand Parvais, Cedric Gustin, Michele Stucchi, Miguel Miranda, Philippe Roussel, Pol Marchal, Satya Munaga, Staf Verhaegen, Stefaan Decoutere en al wie ik hier nu vergeet. Ook aan iedereen die betrokken was bij het organiseren van een van de vele interessante workshops en seminaries: bedankt.

Toen ik in oktober 2004 begon, waren de eerste stenen van het onderzoek naar betere SRAM geheugens reeds gelegd. Bedankt Evelyn, Hua Wang en Peter om de heetste kastanjes uit het vuur te halen.

I would also like to thank Jos Huisken and Maryam Ashouei and the other team members from IMEC Holst for providing the opportunity to integrate our memories in a full system. Your enthusiasm is contagious. Thanks to Vibhu for working on the actual realisation.

Also thanks to Joel Phillips, our cooperation was a very interesting experience. Thanks as well to all the others there at Cadence Berkeley Labs back then for showing me the safer options for lunch in the neighbourhood and thanks for the uplifting discussions.

De 91.33 was niet alleen het lokaal met het raam zonder uitzicht, gekenmerkt door de sporadische benzinegeur en het zalvende gezoem van resonerende lampenkappen zonder reflecterende achterkant. Het was ook ons kleine eilandje binnen ESAT. Eerst veilig afgeschermd van de rest van MICAS met een veilig-

heidsdeur, na het verplaatsen van de deur net dat tikkeltje minder xenofoob. Zonder Anselme, Bram, Jorg, Junfeng, Kristof, Nele, Peter en Vibhu was het niet hetzelfde geweest. Een extra woordje van dank hier voor Jorg, Kristof en Peter voor de bijstand tijdens de laatste loodjes van de eerste tape-out – het was nodig. Ook een extra woordje van dank voor Bram en Nele. Hun enthousiasme en doorgedreven correctiewerk hebben zeker bijgedragen tot het niveau van deze tekst. And Anselme: thanks for the coffee breaks / brainstorm sessions.

Het opstellen en geven van oefenzittingen voor het vak Elektronische Basisschakelingen, h01m3 in de volksmond, was erg leerrijk. Zowel Willy Sansen als Michiel Steyaert hebben me altijd erg vrij gelaten in de invulling hiervan, en toch hebben ze het feit dat de studenten niet altijd even goed presteerden op het examen nooit aan mij geweten - waarvoor dank. De samenwerking met Brecht, Hagen en Pieter verliep altijd bijzonder aangenaam en vlot.

De laatste jaren hebben we er meermaals over gedacht om de kennisoverdracht binnen MICAS een duwtje in de rug te geven. Het is dankzij Cedric Walraevens, Johan Loeckx, Peter Dedene en Pieter Palmers dat daar uiteindelijk ook iets van in huis is gekomen. Ook een klein applausje voor zij die vervolgens informatie aan de MICAS Wiki hebben toegevoegd. Ook een groet hier aan alle collega's waarmee het aangenaam vertoeven was, in de gang, aan de koffietafel, ergens onderweg of gewoon in hun bureau, al dan niet bij nacht en ontij.

De voorbije jaren waren ook buiten de werksfeer bijzonder. Zoveel mensen, plaatsen en gebeurtenissen die we nooit zullen vergeten. Reizen in uitgelezen gezelschap: Berlijn, Costa Rica, Spanje, Finland, San Diego, Boston en Londen; Sangria in het Park; kaartavonden; optredens; nachtelijke filosofische en anderssoortige discussies; shoppen met een heerlijk enthousiaste Roos; middageten dan wel brunchen met Robby en de waanzin; rustige avondjes, al dan niet belaagd door een nogal territoriaal ingesteld konijn; gitaarles; het redden van een zus in nood; wonen in een duiventil met kunst aan het plafond; een huisje kopen en nog zoveel meer. Bedankt Sig, Jorg, Eefje, Michael, Cedric, Roos, Tom, Kris, Robby, Saskia, An, Geert, Annemie, Brecht, Hanna, Johan, Yves, Jan Frederik, Liesbet, Kaat, Bavo en Tinne, jullie maken het verschil.

En dan - "broers", zoals de jongste zeggen zou: Dirk, Johan en Tom, merci om er te zijn zoals jullie zijn. En bedankt papa, bedankt mama, voor jullie toewijding en steun. Tenslotte bovenal een heel oprechte dankejewel en een zoen voor Virginie, die me liefheeft en steunt, ook op de minder opgewekte dagen.

Stefan

Leuven, Mei 2009

Abstract

Variability-aware design of low power SRAM memories

A low energy consumption per operation is one of the most important requirements for the design of portable devices such as smart phones. Promising new applications like intelligent sensors for Ambient Intelligence can only be realised if a significant further reduction in the energy consumption per operation is achieved. In most applications, memories consume a significant part of the energy budget. After optimization, typical applications have strong data locality, hence the small memories that are closest to the processor often dominate the energy consumption.

This work introduces novel, more energy efficient circuits and memory organisations for those relatively small SRAM memories with a moderate speed. The new challenges that are introduced by deep sub-micron technologies are explicitly addressed: leakage currents, large, uncorrelated variations between transistors within one same chip and the dominant impact of interconnect wires on performance.

The most important innovations are an improved bit line structure, the use of dynamic read stability for the memory cell, calibrated global read sense amplifiers and a more efficient implementation of memories with multiple speed modes, thanks to calibrated timing and selective supply voltage scaling. This text discusses all important components of SRAM memories and positions the new techniques in a broad overview of the existing techniques.

The new techniques are illustrated and validated with two successful prototype memories. The most recent prototype is a memory with $4K$ words of 32 bits ($128Kbit$) in a $90nm$ technology. It offers several operation modes. In the $1.2V$ mode, the maximal clock frequency is $850MHz$ and the energy per operation is $8.4pJ$. In the $0.6V$ mode, the maximal clock frequency is $240MHz$ and the energy consumption is $2.7pJ$. This prototype has a significantly lower energy consumption per operation than other published memories with a speed between $100MHz$ and $850MHz$.

Variabiliteitsbewust ontwerp van SRAM geheugens met een zeer laag energieverbruik

Een laag energieverbruik per bewerking is één van de belangrijkste vereisten bij het ontwerp van draagbare toestellen zoals smartphones. Veelbelovende nieuwe toepassingen zoals intelligente sensoren voor Ambient Intelligence kunnen maar worden verwezenlijkt als het energieverbruik per bewerking nog aanzienlijk daalt. In de meeste toepassingen nemen geheugens een belangrijke plaats in in het totale energieverbruik. Na optimalisatie vertonen typische applicaties een sterke datalokaliteit, zodat het energieverbruik van de relatief kleine geheugens die dicht bij de processor staan vaak dominant is.

Dit werk introduceert nieuwe energiezuinigere schakelingen en geheugenorganisaties voor deze relatief kleine SRAM geheugens met een gematigde snelheid. Hierbij wordt expliciet rekening gehouden met de nieuwe uitdagingen die samengaan met de hedendaagse technologieën: lekstromen, grote ongecorreleerde variaties tussen transistoren binnen één chip en de dominante impact van de verbindingsdraden op de performantie.

De belangrijkste innovaties zijn een verbeterde bitlijnstructuur, het gebruik van dynamische leesstabiliteit voor de geheugencel, gekalibreerde leesversterkers en een efficiëntere implementatie van geheugens met meerdere snelheidsmodes dankzij het gebruik van gekalibreerde timing en het selectief verlagen van voedingsspanningen. Deze tekst bespreekt alle belangrijke onderdelen van een SRAM geheugen en kadert de nieuwe technieken in een breed literatuuroverzicht.

De nieuwe technieken worden geïllustreerd en gevalideerd met twee succesvolle prototype geheugens. Het meest recente prototype is een geheugen met $4K$ woorden van 32 bits ($128Kbit$) in een $90nm$ technologie. Het ondersteunt verschillende werkingsmodes. In de $1.2V$ mode is de maximale kloksnelheid $850MHz$ en het totale energieverbruik per operatie is $8.4pJ$. In de $0.6V$ mode is de maximale kloksnelheid $240MHz$ en het totale energieverbruik per operatie is $2.7pJ$. Dit prototype heeft een aanzienlijk lager energieverbruik per operatie dan de andere geheugenontwerpen uit de literatuur voor snelheden tussen $100MHz$ en $850MHz$.

Nederlandse samenvatting

Variabiliteitsbewust ontwerp van SRAM geheugens met een zeer laag energieverbruik

Inleiding

Draagbare elektronische toestellen zijn niet meer weg te denken uit het dagelijkse leven, denk maar aan draagbare muziekspelers, videospelers, spelconsoles en smartphones. Consumenten verwachten almaar indrukwekkendere prestaties van deze draagbare toestellen, terwijl de batterij die deze prestaties moet ondersteunen amper krachtiger wordt. Een laag energieverbruik per functie is dan ook één van de belangrijkste vereisten bij het ontwerp van dit type toestellen. Deze toestellen werken over het algemeen met een kloksnelheid tussen $100MHz$ en $1GHz$.

Het belang van een laag energieverbruik per operatie is niet beperkt tot deze klassieke draagbare toestellen. Ook in hoogperformante systemen zoals krachtige desktop processoren en digitale signaalverwerkingsprocessoren (DSPs) is een laag energieverbruik zeer belangrijk. Een lager energieverbruik maakt het voor de fabrikant mogelijk om goedkopere verpakkingen en koelsystemen te gebruiken [Cla01] en beperkt de energiefactuur voor de klant. De kloksnelheid voor deze toestellen ligt tegenwoordig typisch tussen $500MHz$ en $4GHz$.

Een relatief nieuw toepassingsdomein waarvoor een laag energieverbruik van het grootste belang is, is Ambient Intelligence [Bas03]. Vooral de implementatie van de zogenaamde microWatt sensor nodes stelt nieuwe uitdagingen. Om de droom van Ambient Intelligence volledig waar te kunnen maken, moeten de microWatt nodes jarenlang autonoom blijven werken. Zelfs met geavanceerde energy scavenging systemen, waarbij de microWatt node de energie die in de omgeving aanwezig is zelf omzet in nuttige elektrische energie, kunnen deze systemen slechts over een gemiddeld vermogen van ongeveer $100\mu W$ per cm^2 beschikken [Dec05]. De vereiste klokfrequentie loopt sterk uiteen afhankelijk van de functie van de sensor in kwestie. Een sensor node die dient om de druk in een oog op te volgen, kan volstaan met om de 10 minuten 2000 instructies uit te voeren [Seo08], wat neerkomt op 3 operaties per seconde. Sensor nodes die sneller veranderende data zoals een hartslag moeten opvolgen, hebben meer rekenkracht nodig. Indien deze sensor nodes de signalen ook lokaal willen interpreteren, bijvoorbeeld om de hoeveelheid communicatie te beperken, zijn er al snel enkele miljoenen operaties per seconde nodig [DN07].

In de meeste van deze toepassingen nemen geheugens een belangrijke plaats in in het totale energieverbruik [Ver07, Lam05]. Dat geldt zeker voor de hedendaagse multimedia toepassingen die erg data-intensief zijn [Mas02], maar evengoed voor de laagvermogen sensor nodes [DN07]. Na optimalisatie vertonen de meeste applicaties een sterke datalokaliteit, zodat het energieverbruik van de relatief kleine geheugens die dicht bij de processor staan vaak dominant is. Energiezuinige implementaties voor deze geheugens zijn dan ook onmisbaar om de mogelijkheden van deze toestellen verder te kunnen uitbreiden.

Tot voor kort verbeterde het energieverbruik per operatie van zowel de geheugens als de processoren met de regelmaat van de klok dankzij de voortschrijdende miniaturisatie van de productietechnologie. Kleinere minimale afmetingen lieten niet alleen toe om meer transistoren op dezelfde oppervlakte te krijgen, maar maakten ook hogere snelheden en een lager energieverbruik per operatie mogelijk. In deze periode van "happy scaling" [Dec05] daalde de energie per operatie evenredig met λ^3, waarbij λ de kleinste afmeting is die in de technologie verwezenlijkt kan worden. Wanneer een ontwerp uit een $0.5\mu m$ technologie uit 1990 zonder verdere wijzigingen herschaald wordt naar een $130nm$ technologie uit 2003, daalt het energieverbruik per operatie met een factor 50, zonder significante inspanning van de ontwerper.

Met de invoering van de $90nm$ technologie node kwam dat tijdperk van "happy scaling" tot een eind [Dec05, Gie05]. Doordat lekstromen dominant werden in het energiebudget, kon de drempelspanning niet langer verlaagd worden. Hierdoor werd het onmogelijk de voedingsspanning nog significant verder te verlagen zonder zwaar aan snelheid in te boeten. De variaties tussen nabijgelegen transistoren, ook gekend als mismatch, stijgen sterk met kleiner wordende afmetingen. Deze mismatch heeft een bijzonder nefaste impact op de beschikbare ruismarges voor geheugencellen [Bha01], wat het verlagen van de voedingsspanning verder bemoeilijkt. De performantie van de verbindingsdraden schaalt minder goed dan die van de transistoren [Bam04]. Er moet nu dan ook al van bij de start van het ontwerp rekening mee gehouden worden.

In de praktijk blijken hoogperformante systemen nog steeds voordeel te halen uit verdere miniaturisatie, alvast tot $45nm$. De energie per operatie daalt nu echter in het beste geval evenredig met λ, in plaats van λ^3. Voor systemen met minder activiteit is het eventuele voordeel nog minder duidelijk omwille van het grotere belang van lekstromen.

Om de hoger vermelde nieuwe toepassingen mogelijk te maken, is een verdere verregaande verlaging van de energie per operatie vereist. Dit werk kadert in deze doelstelling. Nieuwe circuit-niveau implementaties voor SRAM geheugens met een lager energieverbruik per operatie worden uitgewerkt. Hierbij wordt actief naar oplossingen gezocht voor de belangrijkste nieuwe uitdagingen: de grote niet-gecorreleerde variaties binnen een chip, de lekstromen en het toenemende belang van de verbindingsdraden. Daartoe worden alle onderdelen van het geheugen en hun onderlinge interacties opnieuw onder de loep genomen in deze tekst.

Geheugencel

De SRAM geheugencel is een van de meest cruciale bouwstenen van een geheugen. In de eerste plaats moet de cel uiteraard functioneel zijn: hij moet 1 bit informatie bewaren, en deze informatie moet uitgelezen en overschreven kunnen worden. De meeste ontwerpen gebruiken een geheugencel met 6 transistoren, een 6T SRAM cel (figuur 1.2). De data wordt bewaard in een bistabiel element dat bestaat uit twee kruisgekoppelde invertoren. Het lezen en schrijven gebeurt via twee toegangstransistoren.

Bij het ontwerp van de geheugencel moet aan veel verschillende specificaties aandacht besteedt worden. De oppervlakte van de cel draagt in belangrijke mate bij tot de totale oppervlakte van het systeem en dus tot de kost. Grotere cellen resulteren ook in langere draden. Langere draden hebben een hogere capaciteit, wat het energieverbruik van het geheugen verhoogt en de snelheid verlaagt. Ook als cellen niet actief worden aangesproken, vloeien er lekstromen door hun transistoren. Dit lekvermogen heeft in veel systemen een nefaste invloed op het totale energieverbruik. De leesvertraging van de cel is vaak een belangrijk onderdeel van de totale toegangstijd van het geheugen. De geheugencel heeft ook een belangrijke directe invloed op het dynamisch energieverbruik van het geheugen via de bitlijn capaciteit. Aangezien de transistorparameters stochastische verdelingen volgen, heeft elk celontwerp een zekere kans om te falen. De ontwerper dient de cel zo te ontwerpen dat deze faalkans laag genoeg ligt, bijvoorbeeld 1 falende cel op 10^9. Dit vereist een betrouwbare methode voor het bepalen van de faalkans.

De marges die traditioneel voor de analyse van cellen gebruikt worden, kunnen ook voor deze statistische analyse worden aangewend. Dit zijn de statische ruismarge (static noise margin, SNM) voor het bewaren van data (SNM_{hold}) en tegen dataverlies tijdens het lezen (SNM_{read}), de cel leesstroom I_{read} en het schrijf-omslag-punt (write trip point, WTP) voor het schrijven. Bij deze statistische analyse moet wel zeer omzichtig worden omgesprongen met veronderstellingen omtrent het type distributie dat de verschillende marges volgen. Meer efficiënte Monte Carlo simulatietechnieken, zoals importance sampling Monte Carlo kunnen het gebruik van deze veronderstellingen vermoedelijk overbodig maken.

Uit simulatieresultaten blijkt dat in een $90nm$ technologie een cel waarvan alle transistoren een minimale lengte en een hoge drempelspanning (SPHVT) hebben en $200nm$ breed zijn, een faalkans onder 10^{-9} heeft en dus volledig operationeel is. In periodes van inactiviteit gaat data slechts verloren indien de voedingsspanning onder $240mV$ zakt. Lezen en schrijven stelt geen probleem zolang de voedingsspanning boven $650mV$ blijft. Indien lagere voedingsspanningen wenselijk zijn of indien ontworpen wordt in een technologie met grotere transistorvariaties, kunnen er twee paden gevolgd worden. Enerzijds kan een andere celtopologie gebruikt worden. Hierbij zijn vooral de 8T cel met leesbuffer en de 6T cel met enkelzijdig lezen en dubbelzijdig schrijven interessant. Anderzijds kunnen de beschikbare marges voor een normale 6T cel ook aanzien-

lijk verbeterd worden door verschillende spanningen te gebruiken voor lezen en voor schrijven.

Lokale geheugenmatrix

Een lokale geheugenmatrix bestaat uit een matrix van cellen en de bijhorende woordlijn decoders en lokale bitlijn periferie.

De woordlijn decoder vormt het laatste niveau van de decoder. Typisch bestaat dit niveau uit een AND functie van een rij activatiesignaal en een kolom activatiesignaal en een buffer functie die de aldus geselecteerde woordlijn aanstuurt. Verschillende implementaties zijn mogelijk, met sterk uiteenlopende snelheid, oppervlakte en lekstromen.

Zoals hoger vermeld is het mogelijk om de beschikbare marges voor een normale 6T cel aanzienlijk te verbeteren door verschillende spanningen te gebruiken voor de verschillende operaties: lezen, schrijven en data bewaren. Deze techniek wordt duidelijk meer en meer toegepast voor geheugens in technologieën na $65nm$. Deze tekst geeft een overzicht van de mogelijke opties en bespreekt de belangrijke aandachtspunten.

Verschillende implementaties voor de lokale bitlijn structuur en periferie worden vergeleken. Een goede bitlijn structuur kan de invloed van de leesstroom I_{read} van de cel en van de bitlijn capaciteit per cel op het energieverbruik en de vertraging van het geheugen aanzienlijk verkleinen.

Wanneer een korte bitlijn met een leesbuffer gebruikt wordt, is de traditionele ruismarge voor de leesoperatie SNM_{read} te pessimistisch. Deze ruismarge is immers gebaseerd op een statische simulatie, met vastgelegde hoge bitlijn spanning. Een korte bitlijn is echter al sterk ontladen vooraleer de cel kan omslaan. Dit zorgt voor extra stabiliteit. Dit werk introduceerde de dynamische leesstabiliteit om deze stabiliteit te kunnen analyseren en benutten, waardoor onnodige veiligheidsmarges vermeden kunnen worden. Deze tekst introduceert een nieuwe marge voor het analyseren van deze dynamische stabiliteit, SNM_{tran}. Deze kan rechtstreeks vergeleken worden met SNM_{hold} en SNM_{read} en is erg lineair als functie van de transistorparameters. Dit vereenvoudigt de analyses voor variaties aanzienlijk.

Leesversterkers

Een leesversterker (Engels: sense amplifier) is een schakeling die een klein ingangssignaal versterkt tot een normaal signaalniveau. Leesversterkers zijn onmisbaar voor een efficiënte data-overdracht tussen geheugencel en geheugenuitgang bij het lezen, en tussen geheugeningang en geheugencel bij het schrijven.

Elke leesversterker heeft een zekere offset spanning V_{offset}. Ingangssignalen groter dan V_{offset} resulteren in een hoog uitgangssignaal, ingangssignalen kleiner dan V_{offset} resulteren in een laag uitgangssignaal. Zonder transistor variaties zou V_{offset} voor alle leesversterkes $0V$ zijn, maar omwille van deze variaties

is ook V_{offset} een stochastische variabele. Voor een degelijk ontworpen leesversterker is de distributie van V_{offset} ongeveer Normaal verdeeld, met standaarddeviatie $\sigma_{V_{offset}} \sim \frac{A_{\Delta V_T}}{\sqrt{W \cdot L}}$. Hierbij is $A_{\Delta V_T}$ de Pelgrom constante van de technologie, en zijn W en L de breedte en lengte van de ingangstransistoren.

Een leesversterker werkt enkel correct indien het aangelegde ingangssignaal groter is dan V_{offset}. Op basis van de kansverdeling en de toegelaten faalkans fr kan dan de minimaal vereiste ingangsspanning ΔV_{min} berekend worden. Deze is steeds evenredig met $\sigma_{V_{offset}}$, die op zijn beurt evenredig is met $\frac{1}{\sqrt{W \cdot L}}$. Voor normale implementaties van leesversterkers is het dynamisch energieverbruik E_{SA} ongeveer proportioneel met de transistorgrootte $W \cdot L$, en dus geldt $E_{SA} \sim \left(\frac{1}{\Delta V_{min}}\right)^2$. Een kleiner ingangssignaal vereist dus een leesversterker met een kwadratisch groter energieverbruik.

Voor de snelheid en het energieverbruik van het gehele geheugen is het echter belangrijk dat de leesversterkers een voldoend kleine ΔV_{min} toelaten. Bij het lezen is ΔV_{min} de signaalzwaai op de bitlijnen. Deze bitlijnen hebben een grote capaciteit C_{BL}. Het energieverbruik per bitlijn is $E_{BL} = C_{BL} \cdot Vdd \cdot \Delta V_{min}$. In typische geheugens vormt E_{BL} een belangrijke bijdrage tot het totale dynamische energieverbruik. Het signaal op de bitlijn moet worden ontwikkeld door de cel leesstroom I_{read}. De celvertraging T_{cell} is dan minstens $\frac{C_{BL} \cdot \Delta V_{min}}{I_{read}}$, en dit is typisch een belangrijk deel van de totale toegangstijd van het geheugen.

In deze tekst wordt een uitgebreide analyse op basis van simulaties in een $90nm$ technologie uitgevoerd van de verschillende leesversterkers die in de literatuur zijn voorgesteld. De verschillende leesversterkers worden ontworpen voor $\Delta V_{min} = 100mV$. Het energieverbruik van de verschillende implementaties verschilt drastisch. De traditionele drain-input latch-type leesversterker verbruikt het minste energie per operatie: $8.3fJ$ per operatie bij een voedingsspanning van $1V$. Het energieverbruik van de andere leesversterkers uit de literatuur ligt boven de $35fJ$. Er is geen zinnig argument om deze te gebruiken.

In situaties waar thermische ruis belangrijk is en in situaties waar de ingangen geen stroom kunnen leveren, is de drain-input latch-type leesversterker echter niet aangewezen. De pulsed current-source leesversterker die we herintroduceerden in [Cos08] is dan de beste keuze met een energieverbruik van $13.7fJ$.

Indien de toegangstijd van het geheugen niet kritisch is, kan het energieverbruik van de leesversterker verder verlaagd worden door een lagere voedingsspanning te gebruiken. Een drain-input latch-type leesversterker met SPLVT transistoren werkt op $450mV$ en verbruikt dan $1.44fJ$ voor $\Delta V_{min} = 100mV$.

Leesversterkers met kalibratie

De stochastische variaties in transistor parameters geven aanleiding tot de evenredigheid $E_{SA} \sim \left(\frac{1}{\Delta V_{min}}\right)^2$. De belangrijkste variaties worden veroorzaakt door het productieproces en veranderen niet tijdens het gebruik en kunnen dan

ook weggekalibreerd worden. Kalibratie doorbreekt de bovenstaande evenredigheid en laat toe om een erg kleine ΔV_{min} te combineren met een lage E_{SA}. De kalibratie wordt uitgevoerd in een daartoe bestemde kalibratiefase. De kalibratie kan eenmalig uitgevoerd worden tijdens het testen na fabrikatie, maar kan bijvoorbeeld ook telkens bij het opstarten worden uitgevoerd, of telkens wanneer de temperatuur meer dan 5 graden veranderd is. De resultaten van deze kalibratie worden opgeslagen en tijdens de normale operatie aangewend als configuratiebits voor de leesversterker. Elke leesversterker heeft zijn eigen configuratiebits. Deze tekst bespreekt drie vormen van kalibratie voor leesversterkers: symmetrische redundantie, tuning en asymmetrische redundantie.

Symmetrische redundantie werd voorgesteld in [Ver08b]. Elke leesversterker wordt geïmplementeerd als N kleinere leesversterkers, waarvan enkel de beste wordt gebruikt. De kleinere leesversterkers zijn allemaal even groot. Een belangrijk nadeel van redundantie is dat er logica in het kritische pad van het geheugen moet worden ingevoegd om de selectie te verwezenlijken. Dit verhoogt het dynamische energieverbruik en maakt het geheugen trager. In praktijk beperkt dit de zinvolle waardes van N tot ongeveer 4. Indien de energie-kosten voor de selectie verwaarloosd kunnen worden, laat symmetrische redundantie met $N = 4$ een 2.2 keer kleinere ΔV_{min} toe voor eenzelfde E_{SA}.

Tuning werkt met één enkele leesversterker. De configuratiebits wijzigen de offset spanning van deze leesversterker. Een eenvoudige en erg effectieve methode om dit te doen is het aanleggen van een aangepaste referentiespanning. In [Cos08] is deze methode toegepast met $N = 16$ mogelijke niveaus voor de referentiespanning. In theorie laat dit een 16 keer kleinere ΔV_{min} toe voor eenzelfde E_{SA}. In praktijk zijn extra marges nodig voor andere onvolmaaktheden zoals thermische ruis en variaties op de voedingsspanning. Metingen tonen aan dat een reductie met een factor 6 kan bekomen worden met deze opstelling.

Deze tekst behandelt ook asymmetrische redundantie. Ook hier wordt elke leesversterker geïmplementeerd als N kleinere leesversterkers. Hier wordt de grootte van de N leesversterkers echter individueel gekozen, zo dat het gemiddelde energieverbruik van de leesversterkers minimaal is. In praktijk wordt dan bijvoorbeeld een 2σ en een 5.5σ leesversterker voorzien. In ongeveer 95% van de gevallen werkt de 2σ versie, in de overige 5% van de gevallen wordt de 5.5σ versie gebruikt. De totale leesversterker faalt enkel indien beide versies niet werken, wat overeenkomt met de faalkans van een traditioneel 6.1σ ontwerp. Het gemiddelde energieverbruik van de SA met redundantie is slechts 14% van die van een normaal ontwerp:

$$E_{avg} = p_1 \cdot E_1 + p_2 \cdot E_2 = \left(95\% \left(\frac{2}{6.1} \right)^2 + 5\% \left(\frac{5.5}{6.1} \right)^2 \right) E_{SA} = 14.3\% \cdot E_{SA}$$

Omgerekend naar ΔV_{min} laat dit een 2.7 keer kleinere ΔV_{min} toe voor eenzelfde gemiddelde E_{SA}. Voor $N = 4$ is dit al 5.3 keer.

Tabel 5.4 geeft een vollediger overzicht van de verbetering die bekomen kan worden met de verschillende kalibratiemethodes.

Data-overdracht

De overdracht van data tussen de cel en de uitgang van het geheugen tijdens leesbewerkingen en tussen de ingang van het geheugen en de cel tijdens schrijfbewerkingen is bijzonder belangrijk voor het energieverbruik en de snelheid van het geheugen.

Voor het lezen kan geopteerd worden voor een enkelzijdige bitlijnstructuur of voor een complementaire bitlijnstructuur. Elke optie heeft voor en nadelen, en de beste keuze hangt af van de omstandigheden.

De bitlijncapaciteit wordt verminderd door het gebruik van lokale bitlijnen waar telkens bijvoorbeeld slechts 16 cellen aan hangen, en deze lokale bitlijnen te connecteren met de leesversterkers aan de geheugenuitgang via globale bitlijnen. De connectie tussen lokale en globale bitlijn kan gebeuren via een pastransistor of met een leesbuffer. In het verleden werd de optie met leesbuffer enkel gebruikt in combinatie met een volledige signaalzwaai op de lokale bitlijn, wat veel energie verbruikt.

In dit werk werd een nieuwe implementatie voor de bitlijnstructuur uitgewerkt: een korte lokale bitlijn met leesbuffer, die een kleine signaalzwaai ontwikkelt op de verticale en horizontale globale bitlijn. De kleine signaalzwaai wordt gecontroleerd door de globale bitlijn op een lage spanning te brengen voor de operatie start, bijvoorbeeld $0.2V$. De leesbuffer kan deze enkel ontladen, en de spanningszwaai is dan ook gelimiteerd tot deze startspanning. In bitlijn structuren zonder leesbuffer is zo een lage startspanning niet mogelijk omdat dit de cel destabiliseert. De signaalzwaai kan dan enkel gecontroleerd worden door de activatietijd van de woordlijn goed te kiezen, wat erg moeilijk is.

Een analyse toont aan dat het geheugen met deze bitlijn structuur minder hinder ondervindt van de transistor variaties. Voor redelijk grote geheugens is deze bitlijn structuur ook nominaal sneller en is het energieverbruik vergelijkbaar. Indien de lage startspanning met een efficiënte spanningsconvertor wordt opgewekt in plaats van met lineaire regulator, vermindert de nieuwe bitlijn structuur het energieverbruik.

Voor de schrijfbewerking moet er een grote spanningszwaai worden aangelegd aan de lokale bitlijn om een voldoend grote schrijfmarge voor de cel te bekomen. Een grote signaalzwaai op de globale bitlijnen heeft echter een nefaste impact op het totale energieverbruik van het geheugen. De meest effectieve oplossing is om lokaal bij de cel een versterker te plaatsen, die een klein signaal op de globale bitlijnen omzet in een volledige signaal op de lokale bitlijnen.

Globale controlecircuits: decoder en timing

De controlecircuits zijn niet alleen onmisbaar voor de correcte werking van het geheugen, ze dragen ook in belangrijke mate bij tot het energieverbruik van het geheugen.

De decoder is de schakeling die op basis van het aangelegde adres één woord in het geheugen selecteert. Afhankelijk van de implemenatie van de rest van het geheugen, moet de decoder ook selectiesignalen aanleveren voor het lokaal blok, de kolom of de rij. Voor het energieverbruik is vooral de globale organisatie van de decoder en het gebruikte signaaltype op de lange draden van belang. De logische gates zelf zijn minder belangrijk voor het energieverbruik.

De correcte opeenvolging van de verschillende gebeurtenissen tijdens een geheugentoegang moet worden gegarandeerd door de timing controleschakelingen. Voor het ontwerp van de timing controleschakelingen moeten verschillende knopen worden doorgehakt. De controleschakelingen kunnen gedistribueerd geplaatst worden in elk lokaal blok, of de timing kan centraal worden beheerd en dan gecommuniceerd worden naar het actieve lokale blok. Verschillende transities van de systeemklok kunnen gebruikt worden als inputs voor de timing controleschakelingen. De correctheid van de timing van een ontwerp kan op drie verschillende manieren worden gegarandeerd. Dit kan op basis van simulatie-resulaten, maar dit vereist de invoering van grote veiligheidsmarges omdat de vertragingselementen niet op dezelfde manier variëren met procesvariaties, temperatuur en voedingsspanning als de schakelingen die gecontroleerd moeten worden. Dit wordt verholpen door het gebruik van replica-gebaseerde timing. Hierbij worden de vertraginselement vervangen door een kopie van het te controleren circuit. In praktijk is het moeilijk om dit efficiënt te implementeren voor geheugens, onder meer omwille van de kleine signalen op de bitlijnen. Replica-gebaseerde timing is ook niet geschikt om stochastische variaties in transistorparameters binnen een chip op te vangen. Kalibratie-gebaseerde timing lost dit probleem op door gebruik te maken van programmeerbare vertragingselementen. De optimale instelling voor deze vertragingselementen wordt tijdens een afzondelijke kalibratiefase bekomen, en daarna tijdens de normale operatie gebruikt. Dit laat toe om voor elke chip individueel de optimale timing instellingen te gebruiken. Voor geheugens met verschillende werkingsmodes laat dit ook toe om voor elke mode een aangepaste instelling te gebruiken.

Een groot deel van de energie voor de decoder en timing wordt verbruikt in het versturen van signalen over de geheugenmatrix. Het versturen van deze signalen met een verminderde spanningszwaai is een interessante maar nog steeds erg uitdagende mogelijkheid.

Prototypes

Twee prototype geheugens werden succesvol gerealiseerd en uitgemeten. Deze prototypes tonen de haalbaarheid en het nut van de nieuw voorgestelde technieken aan.

Het tweede prototype is een geheugen met $4K$ woorden van 32 bits ($128Kbit$) in een $90nm$ technologie. Het ondersteunt verschillende werkingsmodes. In de $1.2V$ mode is de maximale kloksnelheid $850MHz$ en het totale energieverbruik per operatie is $8.4pJ$. In de $0.6V$ mode is de maximale kloksnelheid $240MHz$

en het totale energieverbruik per operatie is $2.7pJ$. De beperkte invloed van de lagere spanning op de snelheid van het geheugen is te danken aan het selectief verlagen van voedingsspanningen: de cel- een woordlijnspanning worden minder sterk verlaagd dan de spanning voor de andere schakelingen. Deze circuits ondervinden anders het meeste hinder van een verlaagde voedingsspanning, terwijl hun bijdrage aan het totale energieverbruik minder dan 5% bedraagt.

Dit prototype behaalt duidelijk een lager energieverbruik per operatie dan de andere geheugenontwerpen uit de literatuur voor snelheden tussen $100MHz$ en $850MHz$.

Besluit

Geheugens met een lager energieverbruik per bewerking effenen het pad voor interessante nieuwe toepassingen, zowel voor de reeds ingeburgerde draagbare toestellen zoals smart phones als voor relatief nieuwe domeinen zoals intelligente sensoren.

In dit werk werden nieuwe energiezuinigere schakelingen en geheugenorganisaties uitgewerkt voor relatief kleine SRAM geheugens (van enkele Kbit tot enkele Mbit) met een gematigde snelheid ($100MHz$ tot $850MHz$). Hierbij werd expliciet rekening gehouden met de nieuwe uitdagingen die samengaan met de hedendaagse technologieën: lekstromen, grote ongecorreleerde variaties tussen transistoren binnen één chip en de dominante impact van de verbindingsdraden op de performantie.

In deze context werden verscheidene nieuwe technieken uitgewerkt.

- Een nieuwe bitlijn structuur vermindert het energieverbruik en verbetert de snelheid van de data-overdracht tussen cel en geheugenuitgang. De nieuwe structuur gebruikt een korte lokale bitlijn met een leesbuffer, die de globale bitlijn aanstuurt. In tegenstelling tot vroegere ontwerpen, ontwikkelt de leesbuffer een goed gecontroleerde kleine signaalzwaai op de globale bitlijn. Deze structuur verkleint de impact van de cel leesstroom op de geheugensnelheid. Dit laat toe transistoren met een lagere lekstroom te gebruiken in de cellen en verbetert de weerstand tegen variaties in de celtransistoren.

- De korte, afgebufferde bitlijn laat het gebruik van dynamische leesstabiliteit toe, wat de beschikbare celmarges verbetert. In deze tekst werd ook een verbeterde maatstaf voor deze stabiliteit uitgewerkt.

- Verbeterde leesversterkers verbeteren zowel het energieverbruik als de vertraging van het geheugen. In dit werk werd een leesversterker van onder het stof gehaald die minder last heeft van thermische ruis en geen DC stroom aan zijn ingangen onttrekt. Dit werk introduceerde ook twee nieuwe, effectievere kalibratietechnieken voor de leesversterkers: tuning en asymmetrische redundantie.

- Een efficiënte afweging tussen energieverbruik en vertraging tijdens de uitvoering werd mogelijk gemaakt door het gebruik van gekalibreerde timing en het selectief verlagen van voedingsspanningen.

Deze technieken werden geïllustreerd en gevalideerd met twee succesvolle prototypes. Het tweede prototype in een $90nm$ technologie behaalt duidelijk een lager energieverbruik per operatie dan de andere geheugenontwerpen uit de literatuur voor snelheden tussen $100MHz$ en $850MHz$.

Deze tekst bespreekt alle belangrijke onderdelen van een SRAM geheugen en kadert de nieuwe technieken in een breed overzicht van bestaande technieken in de hoop dat dit gerichter toekomstig onderzoek mogelijk maakt.

Contents

Woord vooraf **i**

Abstract **v**

Nederlandse samenvatting **vii**

Contents **xvii**

List of Figures **xxiii**

List of Tables **xxxi**

Symbols and Abbreviations **xxxiii**

1 Introduction **1**

1.1 Context and motivation 1

1.2 Objectives 3

1.3 A brief introduction to SRAM memories 4

1.3.1 Traditional memory organisation 4

1.3.2 The memory cell 5

1.3.3 Improved memory organisations 6

1.4 Contributions to the state of the art 10

1.5 Structure of this text 11

2 Memory cell **15**

2.1 Introduction 15

2.2 Important cell specifications 15

2.2.1 Area 15

2.2.2 Static power consumption 16

2.2.3 Read delay 17

2.2.4 Write delay 18
2.2.5 Read energy consumption 18
2.2.6 Write energy consumption 19
2.2.7 Bit line load 19
2.2.8 Bit line leakage 19
2.2.9 Cell yield 21
2.2.10 Reliability 22
2.3 6T SRAM cell: operation and analysis 23
2.3.1 Data retention 25
2.3.2 Read access 38
2.3.3 Write access 40
2.4 Analysis of the impact of transistor variations 44
2.4.1 Traditional Monte Carlo simulation 44
2.4.2 Faster sample evaluations 47
2.4.3 Response surface models 47
2.4.4 More efficient Monte Carlo methods 51
2.5 Design options for the 6T SRAM cell 53
2.5.1 Accurate distributions for the basic cell 53
2.5.2 Impact of transistor dimensions and threshold voltages . . 58
2.5.3 Impact of cell supplies 62
2.5.4 Impact of bulk voltages 62
2.5.5 Impact of bit line precharge voltage 68
2.5.6 Combined impact of Vdd_{cell} and Vdd_{WL} 68
2.5.7 Practical design options 68
2.6 Layout and wire routing 71
2.7 Alternative cell types 74
2.7.1 8T-SRAM cell (readbuffer) 74
2.7.2 Single-ended cell 77
2.7.3 Single-ended read, double-ended write cell 79
2.7.4 Read-static-noise-margin-free 7T-cell 80
2.7.5 Hysteresis cell 80
2.8 Summary 81

3 Local memory matrix **83**
3.1 Introduction . . . 83
3.2 Local block organisation . . . 84
3.3 Word line drivers . . . 89
3.3.1 Example word line driver . . . 90
3.3.2 Interesting implementations . . . 94
3.3.3 Leakage reduction . . . 97
3.4 Local control periphery to assist cell operation . . . 97
3.4.1 Static power consumption . . . 97
3.4.2 Read stability, read current and write-ability . . . 100
3.5 Bit line periphery and hierarchy . . . 101
3.5.1 Traditional bit line structure . . . 101
3.5.2 Domino evaluate read bit line . . . 103
3.5.3 Hierarchical divided bit line . . . 105
3.5.4 Hierarchical bit line with local SA . . . 107
3.5.5 Per-bit SA scheme . . . 108
3.5.6 Short buffered bit line and shared write receiver . . . 110
3.5.7 Short buffered bit line with write-after-read . . . 112
3.5.8 Overview . . . 113
3.6 Dynamic read stability . . . 115
3.6.1 Concept and proliferation . . . 115
3.6.2 Metrics for dynamic read stability . . . 117
3.6.3 Design options to improve SNM_{tran} . . . 120
3.7 Summary . . . 124

4 Sense Amplifiers **127**
4.1 Introduction . . . 127
4.2 Sense amplifier offset voltage and “noise” sources . . . 128
4.2.1 Offset voltage of strobed SAs . . . 128
4.2.2 Parameters impacting SA offset after fabrication . . . 129
4.2.3 Impact of thermal noise . . . 130
4.2.4 Dynamic offset voltage . . . 132
4.3 Sense amplifier ecosystem . . . 132
4.3.1 Input signal . . . 132

4.3.2 Output load . . . 134
4.4 Important specifications . . . 134
4.4.1 Required input signal . . . 134
4.4.2 Dynamic energy consumption . . . 137
4.4.3 Static energy consumption . . . 137
4.4.4 Delay . . . 137
4.4.5 Area . . . 138
4.5 Analysis of strobed voltage-mode SAs . . . 139
4.5.1 Analysis of the offset distribution . . . 139
4.5.2 Analysis of SA delay . . . 142
4.5.3 Analysis of SA energy . . . 143
4.5.4 Analysis of SA noise behaviour . . . 144
4.6 Example SA implementations . . . 144
4.6.1 Drain-input latch-type SA . . . 144
4.6.2 Gate-input stacked latch-type SA . . . 152
4.6.3 Look-ahead type SA . . . 155
4.6.4 Self-closing SA (design 1) . . . 158
4.6.5 Pulsed current source SA (design 2) . . . 161
4.6.6 Double-tail latch-type SA . . . 164
4.6.7 Noise performance . . . 166
4.6.8 Comparison . . . 168
4.7 Summary . . . 169

5 Sense Amplifier Calibration 171
5.1 Introduction . . . 171
5.2 Limitations of traditional SA designs . . . 171
5.3 SA redundancy . . . 174
5.4 SA tuning . . . 176
5.4.1 Concept . . . 176
5.4.2 Implementation . . . 178
5.4.3 Calibrating the tunable SAs . . . 180
5.5 Asymmetric sense amplifier redundancy . . . 182
5.6 Summary . . . 183

6 Data Transfers **189**
6.1 Introduction . . . 189
6.2 Read . . . 189
6.2.1 Signalling types . . . 189
6.2.2 Overview of solutions . . . 193
6.2.3 Analysis under intra-die variations . . . 199
6.2.4 Precharge control . . . 202
6.3 Write . . . 203
6.3.1 Reduced bit line swing . . . 205
6.3.2 Signalling scheme . . . 206
6.3.3 Charge recycling . . . 208
6.4 Summary . . . 208

7 Global control circuits: decoder and timing **211**
7.1 Introduction . . . 211
7.2 Decoder . . . 211
7.2.1 Decoder topologies . . . 212
7.2.2 Predecoders . . . 216
7.3 Timing . . . 219
7.3.1 Events and their delay constraints . . . 219
7.3.2 Local versus global timing control . . . 221
7.3.3 Reference external events for timing . . . 224
7.3.4 Methods to ensure correct operation order . . . 224
7.4 Signal transmission . . . 226
7.5 Example implementation . . . 229
7.5.1 Decoder . . . 229
7.5.2 Operation and timing . . . 231
7.6 Summary . . . 233

8 Prototypes **235**
8.1 Introduction . . . 235
8.2 First prototype ($0.18\mu m$) . . . 235
8.2.1 Design description . . . 235
8.2.2 Measurement results . . . 238

8.3 Second prototype ($90nm$) 241
8.3.1 Design description 241
8.3.2 Measurement results 245
8.4 Comparison with the State of the Art 248

9 Conclusions **251**
9.1 Summary of the chapters 251
9.2 Conclusions 254
9.3 Future work 256

Appendices

A Practical methods to obtain SA noise performance **259**

B Analytical analysis of strobed voltage-mode SAs **263**

C Important technology parameters **269**

D Consequences of further scaling **273**

E List of Publications and Biography **283**

Bibliography **285**

List of Figures

1.1 The traditional SRAM organisation. 4
1.2 Traditional 6T SRAM cell. 5
1.3 The read operation for the traditional 6T cell. 6
1.4 The write operation for the traditional 6T cell. 7
1.5 A memory organisation with subdivided word lines. 8
1.6 A close-up of a memory organisation with subdivided word lines. 8
1.7 Subdivided bit lines. 9
1.8 Bit line capacitance for subdivided bit lines as function of T . . . 10
1.9 Structure of this text. 12

2.1 The most important leakage paths in a traditional 6T SRAM cell. 16
2.2 10T SRAM cell that reduces the bit line leakage. 20
2.3 Traditional 6T SRAM cell. 24
2.4 The example asymmetric 6T cell. 24
2.5 Traditional 6T cell during data retention. 25
2.6 Setup to simulate SNM_{hold}. 26
2.7 Butterfly curves for SNM_{hold}. 27
2.8 A cross-coupled invertor pair with static noise sources. 28
2.9 Impact of static noise sources on the butterfly curves. 28
2.10 The SNM of an unstable cell is negative. 29
2.11 SNM_{hold} as function of $V_{n,left}$ when $V_{n,right} = 0$. 30
2.12 Separate static noise sources $V_{n,left,N}$ and $V_{n,left,P}$. 30
2.13 The impact on the butterfly curves when $V_{n,left,N}$ and/or $V_{n,left,P}$ are present ($V_{n,right} = 0$). 31
2.14 SNM_{hold} as function of $V_{n,left,N}$ (N) and $V_{n,left,P}$ (P). 32
2.15 Transfer curves of invertors at different supply voltages. 33
2.16 Difference between V_{trip} and the ideal trip voltage of $\frac{Vdd}{2}$. 34
2.17 SNM_{hold} as function of the supply voltage. 34

2.18 Circuit used in the definition of SNM. 35
2.19 Cell leakage currents. 36
2.20 Leakage of the example cell as function of Vdd. 37
2.21 A traditional 6T cell during a typical read access. 38
2.22 The circuit setup to analyse read stability by means of the SNM. 39
2.23 Butterfly curves of the asymmetric example cell for different V_{WL}. 39
2.24 SNM_{read} of the asymmetric example cell as function of V_{WL}. . . 40
2.25 The voltage on the internal node that stores a logical 0. 40
2.26 I_{read} and SNM_{read} as function of the word line voltage. 41
2.27 A traditional 6T cell during a typical write access. 41
2.28 Circuit setup to obtain the write trip point WTP 42
2.29 Simulation of the write trip point of the asymmetric example cell. 43
2.30 WTP as function of word line voltage V_{WL}. 43
2.31 SNM_{read} of the asymmetric example cell as function of $V_{n,left}$. . 44
2.32 Histogram of SNM_{hold}. 46
2.33 Cumulative distribution function of SNM_{hold}. 46
2.34 Experimental CDFs for the asymmetric cell at $1V$ and at $750mV$. 48
2.35 Example response surface $SNM_{hold} = f(V_{n,left}, V_{n,right})$. 50
2.36 The baseline 6T SRAM cell for this section. 53
2.37 Experimental CDF of SNM_{read}. 54
2.38 Experimental CDF of $SNM_{hold,0}$ and SNM_{hold}. 54
2.39 Experimental CDF of WTP_0 and WTP. 55
2.40 Experimental CDF of $I_{read,0}$ and I_{read}. 55
2.41 Experimental CDF of $I_{hold,0}$. 56
2.42 Nominal SNM_{hold} as function of $VddCell$. 57
2.43 Nominal SNM_{read} as function of $VddCell$. 57
2.44 Nominal WTP as function of Vdd. 58
2.45 Impact of Wpd on cell specifications. 59
2.46 Impact of Wpu on cell specifications. 60
2.47 Impact of Lpu on cell specifications. 61
2.48 Impact of $Waccess$ on cell specifications. 63
2.49 Impact of $VddCell$ on cell specifications. 64
2.50 Impact of $VssCell$ on cell specifications. 65
2.51 Impact of $VssBulk$ on cell specifications. 66

2.52 Impact of $VddBulk$ on cell specifications. 67
2.53 Impact of $V_{BL,prech}$ on cell specifications. 69
2.54 SNM_{read}, WTP and I_{read} as function of $VddCell$ and $VddWL$. 70
2.55 Artist's impression of the traditional SRAM layout style. 72
2.56 Artist's impression of the thin cell SRAM layout style. 72
2.57 The 8T SRAM cell with readbuffer. 75
2.58 The 5T-SRAM cell from [Car04]. 76
2.59 The 5T-SRAM cell from [Cos07]. 76
2.60 The single-ended SRAM cell from [Cos08]. 78
2.61 The single-ended cell from [Zha08]. 78
2.62 A cell with single-ended read and double-ended write. 79
2.63 The read-static-noise-margin-free 7T cell [Tak06]. 79
2.64 A Schmitt-trigger based cell [Kul07]. 80
2.65 Another Schmitt-trigger based cell [Kul08]. 81

3.1 A typical floorplan of a local memory block. 84
3.2 Artist's impression layout of a 6T cell with "thinCell" layout. . . 84
3.3 Floorplan of a local memory block with partially subdivided WLs. 86
3.4 Interleaving cells in a memory with fully subdivided WLs. 87
3.5 A basic word line driver. 89
3.6 An example word line driver for a word line with 32 cells. 90
3.7 Typical waveforms for the word line signal at different supply voltages. 91
3.8 Word line driver delay as function of Vdd. 92
3.9 Word line driver leakage currents. 92
3.10 Word line driver leakage currents as function of Vdd. 93
3.11 Word line driver dynamic energy consumption. 94
3.12 Word line driver with shared NMOS for the NAND gate. 95
3.13 Word line driver with a dynamic NAND gate 95
3.14 Two-transistor word line driver. 96
3.15 Replica Access Transistors (RATs) to reduce the word line voltage in a way that more or less tracks the PVT corners. 100
3.16 The traditional bit line scheme. 102
3.17 Bit line structure with single-ended full-swing dynamic ripple evaluation. 104

3.18 Hierarchical divided bit line structure. 106
3.19 Hierarchical bit line structure with local sense amplifier. 107
3.20 Per-bit SA scheme. 109
3.21 The local bit line periphery as discussed in [Tak06]. 110
3.22 A short buffered bit line with shared write receiver [Cos07]. . . . 111
3.23 The local bit slice from [Cos09]. 112
3.24 With short buffered bit lines, a cell can have negative SNM_{read} without suffering from read upset. 116
3.25 Circuit setup including noise sources V_n. 118
3.26 Dynamic stability metrics V_{max} and ΔV_{min} as function of V_n. . . 118
3.27 SNM_{tran} as function of C_{LBL}. $(VddCell = 700mV)$. 119
3.28 SNM_{tran} as function of C_{LBL}. $(VddCell = 1000mV)$. 120
3.29 Empirical Cumulative Distribution Functions. 121
3.30 SNM_{tran} as function of C_{LBL} for different values of Wup. 122
3.31 SNM_{tran} as function of the word line rise time for different values of C_{LBL}. 123
3.32 SNM_{tran} as function of $V_{BL,precharge}$ and C_{LBL}. 123

4.1 Sense amplifier offset voltage V_{offset}. 129
4.2 Strobed sense amplifier with V_{offset} and v_{noise}. 130
4.3 Signalling methods. 133
4.4 V_{offset} as function of the important mismatch parameters. 140
4.5 Schematic of the example SA. 141
4.6 SA delay as function of the important mismatch parameters. . . 142
4.7 SA delay as function of V_{in}. 142
4.8 SA dynamic energy consumption as function of the important mismatch parameters. 143
4.9 SA dynamic energy consumption as function of V_{in}. 144
4.10 The schematic of the drain-input latch-type SA. 145
4.11 The nominal waveforms for the drain-input latch-type SA. 145
4.12 Delay as function of V_{in} for latch-type SA. 147
4.13 Behaviour of SPLVT drain-input latch-type SA as function of Vdd. 149
4.14 Behaviour of SPHVT drain-input latch-type SA as function of Vdd. 150
4.15 Behaviour of the drain-input latch-type SA as function Mtop width. 151
4.16 Schematic of the gate-input stacked latch-type SA. 152

4.17 Nominal waveforms for the gate-input stacked latch-type SA. . . 154
4.18 Schematic of the look-ahead SA. 155
4.19 Nominal waveforms for the look-ahead SA. 157
4.20 Schematic of the self-closing SA. 158
4.21 The schematic of the core part of the self-closing SA. 158
4.22 Nominal waveforms for the self-closing SA. 160
4.23 Schematic of the pulsed current source SA. 161
4.24 Nominal waveforms for the pulsed current source SA. 163
4.25 Schematic of the double-tail latch-type SA. 164
4.26 Example waveforms for the double-tail latch-type SA. 165
4.27 The amplification $A = \frac{v_{out}}{v_{in}}$ as function of time. 167

5.1 Illustration of $n(fr)$. 172
5.2 Relative size of a traditional SA as function of the allowed failure rate fr. 173
5.3 Setup for SA redundancy (N=2). 174
5.4 Energy reduction provided by SA redundancy. 175
5.5 Area reduction provided by SA redundancy. 176
5.6 Effective offset of a tuned sense amplifier. 177
5.7 Setup for SA tuning (N=16 ; word length=32). 178
5.8 Prototype implementation of (a) reference voltage generator, and (b) voltage selector. 179
5.9 Sense amplifier schematic. 179
5.10 Setup during the calibration phase. 181
5.11 Optimal results for asymmetric SA redundancy with $N = 2$. . . . 184
5.12 Optimal results for asymmetric SA redundancy with $N = 3$. . . . 185
5.13 Optimal results for asymmetric SA redundancy with $N = 4$. . . . 186

6.1 Signalling methods. 190
6.2 Delay for discharge with driver in linear region. 192
6.3 Solutions to transfer data from the cell to the memory output. . 194
6.4 Positioning the connectors between vertical and horizontal bit lines. 195
6.5 An example setup with readbuffer (solution C). 197
6.6 Simulation setup to compare extended GBL schemes with and without readbuffer. 199

6.7 Distributions of the delay from WL activation to $150mV$ swing on $HGBL$. 200
6.8 Distributions of the effectively developed swings. 201
6.9 Precharge scheme used in [Yan05]. 202
6.10 Precharge immediately after access. 203
6.11 Mixed precharge scheme. 204
6.12 Traditional write scheme. 204
6.13 Shared write receivers. 206
6.14 Energy consumption of different signalling types. 207
6.15 Charge recycling: an example. 207

7.1 Example decoder topologies. 213
7.2 A simple 3 to 8 predecoder with input address latches. 217
7.3 A 4-input static NAND gate. 217
7.4 Implementations for predecoder gates. 218
7.5 Important events for the memory described in [Cos09]. 220
7.6 An entirely local timing circuit for the local block. 222
7.7 A very global timing circuit for the local block. 222
7.8 A timing solution that combines global and local timing control. 223
7.9 Predecoder gate with negative half-swing pulse output. 227
7.10 Half-swing pulse-mode "AND" gate. 228
7.11 A differential medium-swing receiver [Cos07]. 228
7.12 Reducing the number of active receivers. 229
7.13 Decoder organization for the $128Kbit$ memory from [Cos09]. . . . 230
7.14 Decoder implementation. 231
7.15 Simulated timing diagram. 232
7.16 Inverting tunable delay line. 232
7.17 Pulse generator with tunable width. 232
7.18 Generation of critical timing signals. 233

8.1 The fabricated $64Kbit$ memory chip. 236
8.2 A low-swing distributed decoder. 237
8.3 Measured cell DRV distribution. 239
8.4 Measured distributions for destructive read and for write-ability. 239
8.5 An on-chip digital oscilloscope. 240

8.6 The fabricated $128Kbit$ memory chip. 242
8.7 Global memory organization. 243
8.8 Single-ended memory cell. 244
8.9 Measured cumulative distribution of cell operation margins. . . . 246

A.1 Concepts used in the measurement of the input referred noise. . . 260
A.2 CDF of estimated $\sigma_{v_{source}}$. 261

B.1 Schematic of a differential input pair with precharged output. . . 264
B.2 Small-signal schematic of a differential input pair with precharged output. 266
B.3 The small-signal schematic of a differential input pair with precharged output and cross-coupled NMOS load. 267
B.4 The small-signal schematic for a drain-input latch-type SA. . . . 268

C.1 Transistor drain current for SPLVT transistors. 270
C.2 Setup to include transistor variations in simulations. 270
C.3 Transistor capacitances for a SPHVT NMOS transistor. 271
C.4 Transistor capacitances for a SPHVT PMOS transistor. 271

List of Tables

2.1 Baseline cell parameters for this section. 53

3.1 Overview table for bit line periphery and hierarchy. 114

4.1 Representative values for RMS thermal noise voltage at $T = 300K$. 130
4.2 Required $\Delta V_{min,mismatch}$ for different yield targets and upscale factors S. 136
4.3 Overview table of a linear sensitivity analysis for V_{offset}. 141
4.4 Settings used for drain-input latch-type SA. 146
4.5 Settings used for the gate-input stacked latch-type SA. 153
4.6 linear sensitivity analysis for V_{offset} for gate-input stacked latch-type SA. 153
4.7 Settings used for the look-ahead SA. 156
4.8 Settings used for the self-closing SA. 159
4.9 Settings used for the pulsed current source SA. 162
4.10 Result table for the double-tail latch-type SA. 165
4.11 Results of transient noise simulations for drain-input latch-type SA. 166
4.12 Results of transient noise simulations for pulsed current source SA. 167
4.13 Performance values for different SA types at $Vdd = 1V$. 168

5.1 Required SA size to cope with transistor mismatch. 173
5.2 SA tuning: parameters and specs. 180
5.3 Application of the calibration algorithm. 181
5.4 Overview table of the performance of the calibration techniques. 187

7.1 Switched capacitance for decoder topologies 214

8.1 Memory dimensions. 235
8.2 Measured performance values for the first prototype memory. . . 241

8.3 Memory dimensions for the second prototype. 243
8.4 Measured performance values for the second prototype memory. . 246
8.5 Detailed measured energy consumption for the second prototype. 247
8.6 Comparison with other state-of-the-art designs. 249

C.1 Transistor parameters. 269

D.1 Information compiled from the 2007 edition of the ITRS roadmap. 274
D.2 Guestimates for the threshold voltage mismatch. 275
D.3 Impact of transistor variations on the performance of logic gates. 276
D.4 Scaling of the 6T cell. 278
D.5 Scaling of low-swing data transfers. 279

Symbols and Abbreviations

Symbols

$[a:b]$	Set of integers i for which holds $a \leq i \leq b$
$\Omega/\square$	Ohm per square
$\square$	Square, the unit for $\frac{\texttt{wire length}}{\texttt{wire width}}$.
$\sigma_{\Delta V_T}$	The standard deviation of $V_{T,2} - V_{T,1}$, the difference between the threshold voltages of the transistors in a matched transistor pair.
σ_{V_T}	The standard deviation of V_T. $\sigma_{V_T} = \frac{\sigma_{\Delta V_T}}{\sqrt{2}}$.
$A_{\Delta V_T}$	The Pelgrom constant, a technology parameter that is typically expressed as $[mV \cdot \mu m]$. This value is used to calculate $\sigma_{\Delta V_T} = \frac{A_{\Delta V_T}}{\sqrt{W \cdot L}}$ and $\sigma_{V_T} = \frac{A_{\Delta V_T}}{\sqrt{2}\sqrt{W \cdot L}}$.
A_{pre}	The pre-amplification factor, the amplification from the input to the sensitive nodes before regenerative feedback becomes dominant
C_{BL}	Bit line capacitance
$C_{BL,cell}$	Contribution of a single cell to the bit line capacitance
F	Feature size, e.g. $F = 90nm$ for a 90nm technology.
I_{read}	Cell read current
RT	Signal rise time
S	Sub-threshold slope $[\frac{mV}{decade}]$
S	The upscale factor: the gate area of a transistor as compared to the minimal gate area in the technology: $\frac{W \cdot L}{W_{min} \cdot L_{min}}$
SNM	static noise margin
SNM_{hold}	static noise margin during data retention
SNM_{read}	static noise margin for the read operation (based on DC simulation)
SNM_{tran}	static noise margin for the read operation (based on the actual transient simulation)

Vdd	The power supply (voltage). Often around $1V$ in $90nm$ technologies.
Vss	The negative power supply (voltage). Often $0V$.
V_{trip}	invertor trip voltage
V_T	transistor threshold voltage
WTP	write trip point

Abbrevations

BL	Bit Line
BRD	Block Row Decoder, the predecoder that generates BRS
BRS	Block Row Select, the decoder signal that selects a row of local blocks
CD	Column Decoder, the predecoder that generates CS
CDF	Cumulative Distribution Function
CS	Column Select, the decoder signal that selects a column of local blocks
DRC	Design Rule Checks.
DSM	Deep Sub-Micron.
DDSM	Deep Deep Sub-Micron (or Dangerously Deep Sub-Micron). Typically used for technology nodes beyond 90nm, with significant problems with leakage currents and intra-die variations.
GBL	Global Bit Line
GWL	Global Word Line
HGBL	Horizontal Global Bit Line
HVT	High V_T
IO	Input/Output
LBL	Local Bit Line
LL	Low Leakage
LVT	Low V_T
LWL	Local Word Line
M2	Metal Layer 2, the metal layer which is second-closest to the substrate

OPC	Optical Proximity Correction
PDF	Probability Density Function
PVT	Process, Voltage, Temperature. Used to indicate the variations that are common for nearby transistors. Includes gradual variations due to fabrication, supply voltage variations and die temperature.
SA	Sense Amplifier
SOI	Silicon-On-Insulator
SRAM	Static Random Access Memory
SP	Standard Performance
SVT	Standard V_T
VGBL	Vertical Global Bit Line
$WIBS$	WithIn-Block Select, the decoder signal that selects a word in the activated local block
WIBD	WithIn-Block Decoder, the predecoder that generates $WIBS$
WL	Word Line. (sometimes Word Length, the number of bits per word).

Chapter 1

Introduction

1.1 Context and motivation

Portable electronic devices have become a key part of everyday life. Just think about portable media players, gaming consoles and smartphones. Consumers expect ever more performance and features from these devices, while the battery capacity hardly improves. A low energy consumption per function is hence one of the most fundamental requirements in their design. These devices typically operate with a clock frequency between $100MHz$ and $1GHz$.

The importance of a low energy consumption per operation is however not limited to these classical portable devices. Even in high-performance systems such as powerful desktop processors and digital signal processors (DSPs), energy efficiency is a key issue. A lower energy consumption allows the manufacturer to use less expensive packages and cooling solutions [Cla01] and reduces the energy bill for the customer. Today, the clock frequency for these devices is typically in the range from $500MHz$ to $4GHz$.

A relatively new application domain for which low energy consumption is of key importance, is Ambient Intelligence [Bas03]. Especially the implementation of the so-called microWatt sensor nodes introduces significant new challenges. To truly realise the dream of Ambient Intelligence, these microWatt nodes must be able to operate autonomously for many years. Even with advanced methods of energy scavenging, in which the microWatt nodes themselves convert environmental energy into useful electrical energy, these systems will have to operate with an average energy consumption below $100\mu W$ per cm^2 [Dec05]. The required clock frequency depends strongly on the functionality of the sensor. A sensor node to monitor the pressure inside the eye can get away with waking up once every 10 minutes, execute 2000 instructions and go back to sleep [Seo08], which boils down to 3 operations per second. Sensor nodes that have to monitor faster changing signals such as a heartbeat, need more processing power. In situations where the sensor node itself must interpret these signals, for example to reduce the required amount of communication, several millions of operations per second can be required [DN07].

In most of these applications, memories consume a significant part of the total energy budget [Ver07, Lam05]. This is certainly true for the very data intensive contemporary multimedia applications [Mas02], but similar conclusions apply to low power sensor nodes [DN07]. After optimizations have been applied, most

applications have strong data locality. This increases the relative importance of the small memories that are closest to the processor to such an extent that they often dominate the energy consumption. Hence, energy efficient implementations for these relatively small memories are a key requirement to enable further extensions of the capabilities of these devices.

Until recently, the energy consumption per operation of both memories and processors improved on a regular basis thanks to the ongoing miniaturization of technology. Smaller minimal dimensions not only allowed to put more transistors onto the same area, but also enabled higher speeds and lower energy consumption per operation. In this era of "happy scaling" [Dec05], the energy per operation reduced in proportion to λ^3, with λ the smallest feature size that can be realized in the technology. If a design is rescaled from a $0.5\mu m$ technology (1990) to a $130nm$ technology (2003), the energy consumption per operation is reduced with a factor of 50, without significant effort from the designer.

This era of "happy scaling" ended with the introduction of the $90nm$ technology node [Dec05, Gie05]. Because leakage currents started to dominate the energy budget, the threshold voltage could no longer be reduced. This in turn made it impossible to further reduce the supply voltage without taking a significant performance hit. The variations between adjacent transistors, also known as mismatch, increased quickly with smaller dimensions. This mismatch has a detrimental impact on the available noise margins for memory cells [Bha01], which further impedes the reduction of the power supply.

Another important problem for technology scaling is that the performance of wires does not scale as well as that of transistors [Bam04]. Hence, wires have become a first-order design consideration, especially in memory design.

All in all, it turns out that high-performance systems still benefit from further miniaturization, at least down to the $45nm$ node. However, the energy per operation now reduces proportional to λ at best, rather than λ^3. For systems with a lower activity, the potential advantage is less clear due to the larger impact of leakage currents and transistor variations.

To enable exciting new applications, a further radical reduction of energy consumption per operation is needed. Three tracks can contribute to this energy reduction: improvements in process technology, in system-level design and in circuit-level design.

For some applications, further miniaturisation can account for a part of this reduction, but only if new circuit and system solutions are devised to cope with the increasing variability and leakage currents. Even though scaling seems to become less effective and less affordable for many applications, technology can still have a tremendous impact on system performance by offering new integrated technology options such as efficient inductors, high density capacitors, new transistor architectures and new cell types for volatile and non-volatile data storage. To correctly assess this multitude of possibilities and to use them efficiently requires a deep understanding of their circuit-level implications.

At system-level, co-optimisation of the algorithms and the hardware system can often result in a huge reduction in the energy consumption. The maximal benefits from this approach are obtained when application-domain specific memories are employed. These memories can make the optimal trade-off between the specifications (e.g. energy, delay, area, yield and special features) for the application domain.

The third track to reduce the energy consumption, improving the circuit-level design of memories, is the subject of this thesis.

1.2 Objectives

The main objective of this work is to reduce the total energy consumption per access for relatively small embedded Static Random Access Memories (SRAMs) in current and near-future technology nodes by improving their circuit-level implementation.

This thesis only considers scratchpad memories, not caches. Such scratchpads are often used as the first level of the memory hierarchy in embedded systems, systems-on-chip and DSPs. The contents of scratchpad memories is explicitly managed by the application, as compared to caches whose content is managed by hardware. Only memories whose access completes in a single clock cycle are studied. Target clock frequencies range from $100MHz$ and $1GHz$. This text uses a $4Kword$, $128Kbit$ memory with a word length of 32 bits per word as representative example memory size. A typical organisation for this memory size is 512 by 256 cells. All analyses and examples are provided for a $90nm$ technology. The studied techniques and analysis methods can be applied to a range of memory sizes. Beyond this range, other solutions should be considered. When the memory size is very small, e.g. below 64 by 64 cells ($4KBit$), the required specifications typically change, with more emphasis on speed and less emphasis on area and leakage power and the ratio of area of the peripheral circuits to that of the memory cells changes. When the memory size is larger than about 1024 by 1024 cells, the emphasis on area and leakage power increases. For high-speed implementations, the wire delay can become problematic. Most larger memories also feature a pipelined access, with a latency of several clock cycles. Although many of the ideas and analyses that are discussed in this work can be very useful for these very small and these rather large memories, they are not treated explicitly in this work.

This work assumes that the memory is accessed each clock cycle. The considered energy per access is hence the sum of the dynamic energy consumption and the leakage power integrated over one clock period. When correctly designed, the energy consumption per access for the target memory specifications is then dominated by the dynamic energy consumption, not by the leakage power (e.g. [Cos09]). Leakage would become dominant if the system is often idle and no appropriate leakage reduction techniques are employed.

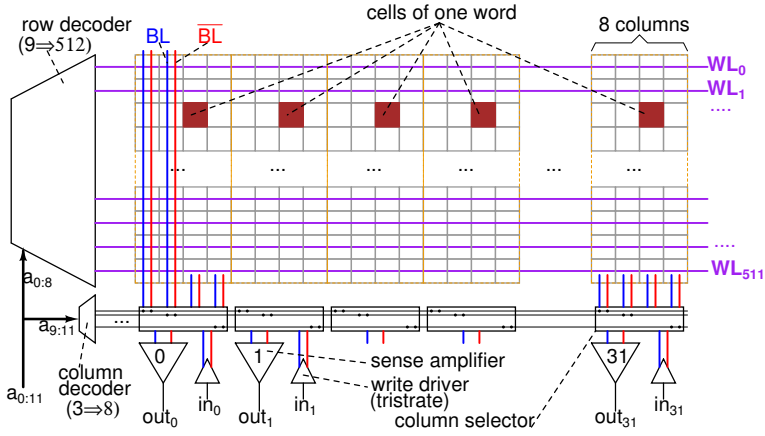


Figure 1.1: The traditional SRAM organisation. The example memory contains 4096 words of 32 bits ($128Kbit$). The cells are organised in 512 rows by 256 columns.

This work explicitly addresses the major new challenges that are introduced by Deep Sub-Micron (DSM) technologies: large uncorrelated intra-die variations, leakage currents and wire dominance. Reliability issues due to time-varying effects are considered to a lesser extent.

As a second objective, this work should level the field for the future design of application domain specific memories and for the analysis of new transistor architectures and cell types.

1.3 A brief introduction to SRAM memories

1.3.1 Traditional memory organisation

Figure 1.1 shows the traditional organisation for small SRAMs. The example memory contains 4096 words of 32 bits ($128Kbit$). The cells are organised in a single matrix of 512 rows by 256 columns. In this text, the word lines WL_i always run horizontally. In the traditional organisation, all cells in one row connect to a single word line. The bit lines BL and $\overline{BL}$ (sometimes referred to as $BLBar$) run vertically. In the traditional organisation, all cells in one column connect to a single bit line pair. The total capacitance of the bit line is called C_{BL}. This capacitance has a large impact on the energy consumption and the read delay of the memory. To limit this impact, low-swing signals are used on the bit lines during read operations. These signals are then detected by the sense amplifiers.

In the example, each word line connects to 256 cells, while the word length is 32. Each word line hence activates 8 different words. Typically, the cells of the different words are interleaved as show in the figure. This way, the column selectors only have to select one out of 8 adjacent bit line pairs to connect to the

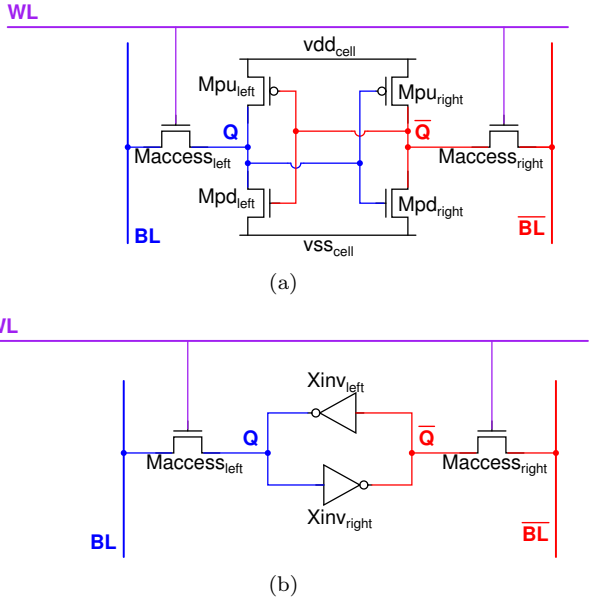


Figure 1.2: Traditional 6T SRAM cell. (a) Transistor schematic. Maccess=access transistor, Mpd=pull-down transistor, Mpu=pull-up transistor. (b) Same cell, different view.

IO circuits, the sense amplifiers and write drivers. In this traditional memory organisation, only one set of 32 sense amplifiers is needed.

Twelve address bits $a_{0:11}$ are required to select one word out of 4096 words ($2^{12} = 4096$). The row decoder decodes 9 address bits and activates one of the 512 word lines. The remaining 3 address bits are decoded by the column decoder to select which of each 8 bit lines must be connected to the sense amplifiers or write drivers by the column selectors. The column selector acts as a multiplexer during write and as an analog demultiplexer during read.

1.3.2 The memory cell

Figure 1.2 shows the schematic of a traditional 6T SRAM cell. While the cell is not being accessed, the word line (WL) is forced to a low voltage which disables the access transistors. The cross-coupled invertors form a latch, a bistable circuit that retains the cell data.

Figure 1.3 shows a typical read operation. Before the word line is enabled, the bit lines BL and $\overline{BL}$ are precharged high. The word line is then enabled and the access transistor at the cell side that stores a "0" conducts a current I_{read}. This current discharges one of the bit lines. The other bit line stays at the precharge voltage. When the word line is activated during a time period $T_{develop}$, a voltage

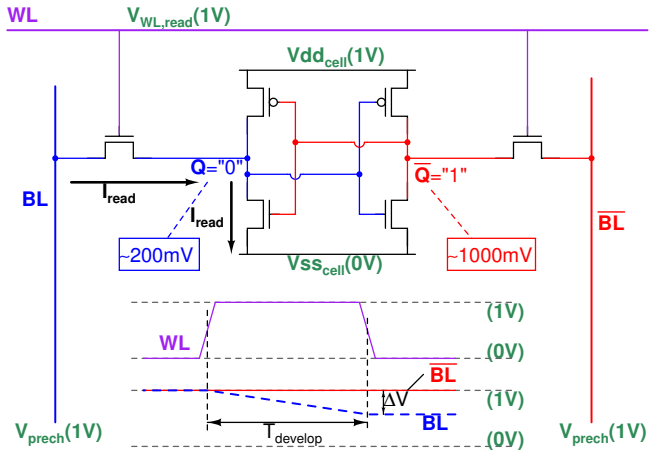


Figure 1.3: The read operation for the traditional 6T cell. The indicated voltages are typical for a cell in a $90nm$ technology.

difference ΔV is developed between the bit lines:

$$\Delta V = \frac{T_{develop} \cdot I_{read}}{C_{BL}}$$

This ΔV is the input signal that is available to the next circuit stage, the sense amplifiers. Conversely, if a voltage difference ΔV is needed, the time $T_{develop}$ must be at least

$$T_{develop} = \frac{C_{BL} \cdot \Delta V}{I_{read}}$$

In typical memory designs, $T_{develop}$ is a major contributor to the total memory access time.

Figure 1.4 shows a typical write operation. One bit line is driven high, the complementary bit line is driven low, typically to $0V$. When the word line is activated, the cell content is overwritten.

1.3.3 Improved memory organisations

The traditional memory organisation from section 1.3.1 has three important disadvantages. **First**, a lot of energy is wasted in the bit lines. Each cycle, 256 bit line pairs are precharged and evaluated, while only 32 of them are actually used. **Second**, there are half-selected cells during a write operation. Their word line is activated, but their bit lines remain precharged high. In traditional operation schemes, these cells are subject to the same conditions as during a read operation, hence there is no problem. However, these half-selected cells significantly limit the options that are available to implement write assist schemes. **Third**, the word line is a long thin wire with a large total capacitive load. For larger matrices, the word line delay becomes dominated by

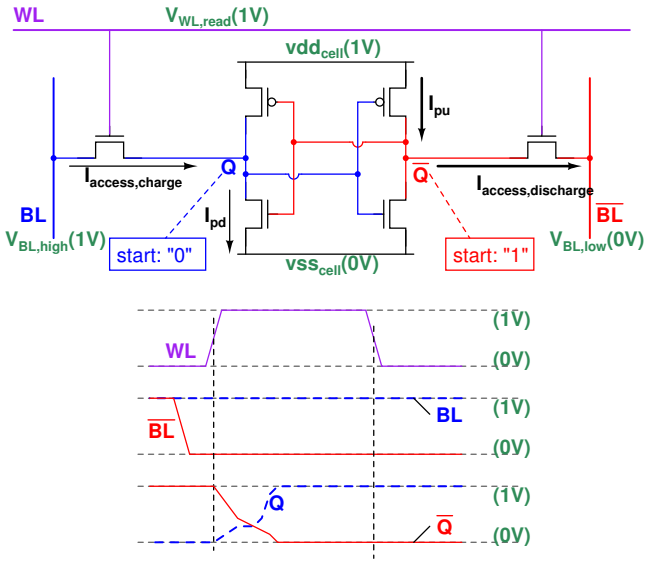


Figure 1.4: The write operation for the traditional 6T cell. The indicated voltages are typical for a cell in a $90nm$ technology.

the wire resistance. The wire delay for a wire with distributed load is given by $\tau = \frac{R_{wire} \cdot C_{load}}{2}$ [Rab03]. In $90nm$, R_{wire} of the word line is about 1.5Ω per cell, and C_{load} is about $1fF$ per cell. If there are K cells connected to a word line, the word line wire delay is

$$\tau = \frac{(K \cdot 1.5\Omega) \cdot (K \cdot 1fF)}{2} = K^2 \cdot 0.75fs$$

For $K = 256$, τ is $49ps$. For $K = 512$, τ is already $196ps$. C_{load} per cell scales as $\sim \lambda$ and R_{wire} per cell scales as $\sim \frac{1}{\lambda}$, so τ is expected to remain approximately constant with technology scaling. As gate delay still improves with technology scaling, the importance of the word line resistance will increase further. Designs in older technologies used poly word lines, which typically have a 100x larger resistance per cell than metal wires.

1.3.3.1 Subdivided word lines

These problems are addressed by using subdivided word lines as shown in figure 1.5 and 1.6. Such divided word line approaches were already considered common for large memory matrices in [Ito95]. The row decoder now activates a global word line GWL. The column select signals CS are now also routed vertically in the corresponding column. The last decoder stage consists of word line drivers that combine GWL and CS to select the correct local word line LWL that connects to the memory cells. In this example, all cells that connect to a single LWL belong to the same word, hence all issues regarding half-selected cells are avoided. In this text, this is called a fully subdivided word line.

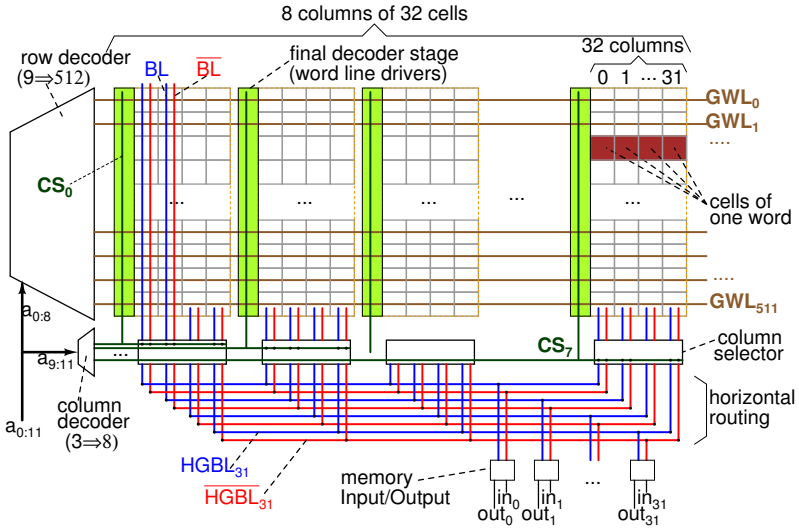


Figure 1.5: A memory organisation with subdivided word lines. The example memory contains 4096 words of 32 bits ($128Kbit$). The cells are organised in 512 rows by 256 columns.

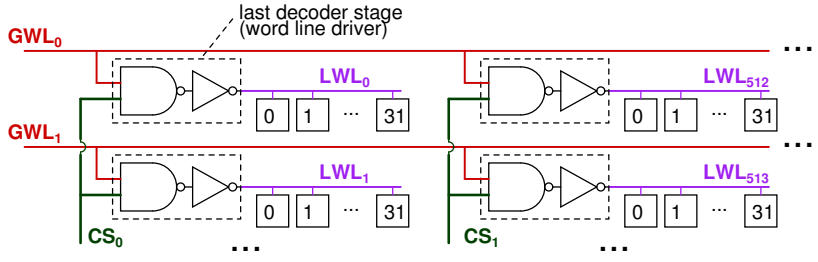


Figure 1.6: A close-up of a memory organisation with subdivided word lines.

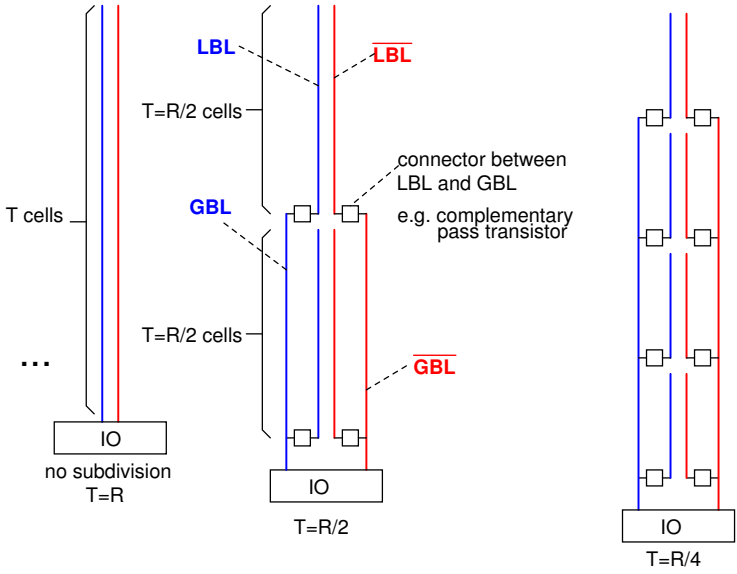


Figure 1.7: Subdivided bit lines.

When subdivided word lines are used, the cells that correspond to the i-th bit of each word are no longer all adjacent, hence horizontal routing is needed in all circumstances. In this text, the wires for horizontal routing are called the horizontal global bit lines ($HGBL$).

Fully subdivided bit lines significantly reduce the energy consumption in the bit lines as now only the required bit lines are evaluated. However, C_{BL} is still large, which has two detrimental consequences. First, the energy consumption in the bit lines is large, especially for write operations because they require a full-swing signal on the bit lines. Second, the cell read delay $T_{develop}$ is large because C_{BL} must be discharged by a small memory cell.

1.3.3.2 Subdivided bit lines

These two problems are addressed by subdividing the bit lines as well. Figure 1.7 shows possible bit line subdivisions. The number of cells per local bit line LBL is called T and there are R rows in the matrix. Figure 1.8 shows how the bit line capacitances change with T for $R = 512$ in a $90nm$ technology. This figure assumes that each cell that connects to a LBL contributes $C_{LBL,cell} = 0.5fF$ to its capacitance, while the wire capacitance per cell of the GBL is $C_{GBL} = 0.15fF$ per cell. Even when the overhead that is added by the connectors between GBL and LBL is included, the total bit line capacitance can be reduced from $256fF$ to $117fF$ (2.2x).

For the write operation, this is already a very significant improvement. The impact on the energy and delay of the read operation depends strongly on the

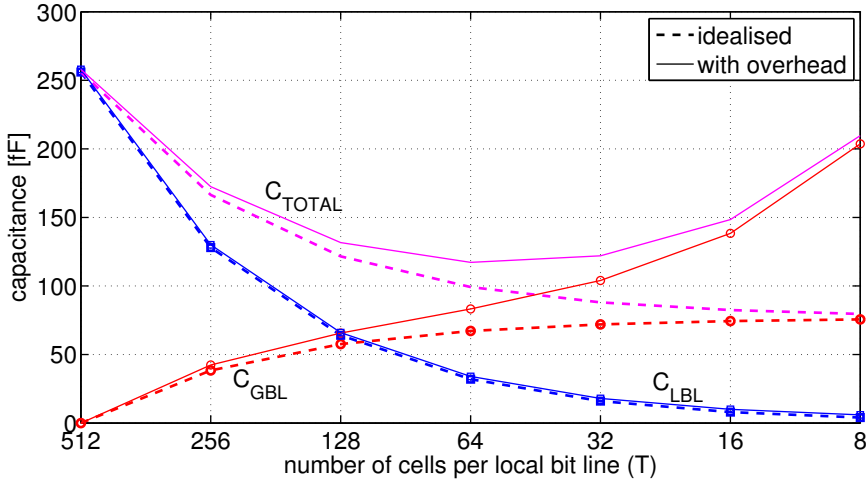


Figure 1.8: Bit line capacitance for subdivided bit lines as function of T for a matrix with $R = 512$ rows. Idealised curves assume that the additional BL periphery does not add any capacitance. The curves that include overhead assume that the local bit line periphery adds $2fF$ to C_{LBL}, and that each connector between GBL and LBL adds $2fF$ to C_{GBL}. ($C_{LBL,cell} = 0.5fF$, $C_{GBL,cell} = 0.15fF$).

implementation that is used to connect the local and global bit lines. Typically, the swing on the LBL is larger than that on the GBL, which reduces the gain in read energy and delay. Still, significant improvements are possible. The write energy can be reduced further by employing a local write receiver. This allows to use a low-swing signal on the GBL for write. The write receiver, for example a local sense amplifier, amplifies this signal on the LBL [Yan05].

1.4 Contributions to the state of the art

This text is structured as a broad yet in-depth overview of the current state of the art regarding the variability-aware circuit design of embedded SRAM memories with a low energy consumption per operation. The improvements that have been introduced during this PhD are positioned at the appropriate location within this overview. This has both its merits and its disadvantages.

On the positive side, the resulting text might actually be useful. The literature regarding SRAMs is mainly focused on result reporting, and no overview texts on the variability-aware design of low-power SRAM memories are available. This text attempts to fill part of this empty space. It might hence be useful for those who are new to the field, as they get access to pre-digested knowledge, which should ease the learning curve. It might be useful as well for those well versed in SRAM design, as they now get an organised overview of all the things they already now, allowing for more targeted discussions and future research.

It might also proof useful for my weary brain, as it can finally forget all that is now written down.

On the down side, this text organisation might make it harder for the reader to judge the contributions of this PhD to the state of the art. To ameliorate this problem to some extent, a short overview of the contributions is provided in this section, along with pointers to the corresponding sections in the text.

Short buffered bit lines with low-swing extended GBL reduce the dynamic energy consumption and improve the speed of the data transfer from memory cell to memory IO (section 6.2.2.3). This bit line structure improves the resilience against intra-die variations (section 6.2.3).

This bit line structure also allows the use of **dynamic read stability** as a method to prevent destruction of cell data during read. The concept of dynamic read stability is treated in section 3.6. A useful **metric for dynamic read stability**, SNM_{tran}, is introduced in section 3.6.2.

Improved strobed sense amplifiers further reduce the dynamic energy consumption and improve the speed of the data transfer from memory cell to memory IO. Section 4.6 shows clearly that the traditional drain-input latch-type sense amplifier has the best energy-delay performance for a specified offset. However, in cases where the noise performance is dominant or in cases where input currents cannot be tolerated, the **pulsed current source sense amplifier** that we re-introduced is the best choice. The energy-offset performance of normal strobed sense amplifiers is fundamentally limited by Pelgrom's law. Chapter 5 discusses how calibration circumvents this limitation. It also shows that **sense amplifier tuning** is more effective than previous implementations. Section 5.5 discloses **asymmetrical sense amplifier redundancy**, another efficient calibration technique.

These techniques have been used in two prototypes, as discussed in chapter 8. The comparison in section 8.4 shows that these prototypes outperform other state-of-the-art memories when a low energy per access must be combined with an access frequency above $100MHz$. The second prototype provides a range of frequency settings. Energy-efficient operation over a wide frequency range from $850MHz$ to $240MHz$ is obtained by a combination of **calibration-based timing** (see section 7.3.4) and **selective voltage scaling** (see [Cos09]).

1.5 Structure of this text

Figure 1.9 shows an overview of the contents of this text.

Chapter 2 addresses the core building block of the SRAM memory, the memory cell. The important cell specifications are outlined, the traditional analysis methods are presented and an overview of methods to analyse the behaviour of SRAM cells under transistor mismatch is provided. Then, the design options for the traditional 6T SRAM cell are explored, with special attention to the impact of bias voltages on cell stability and write-ability. A brief introduction

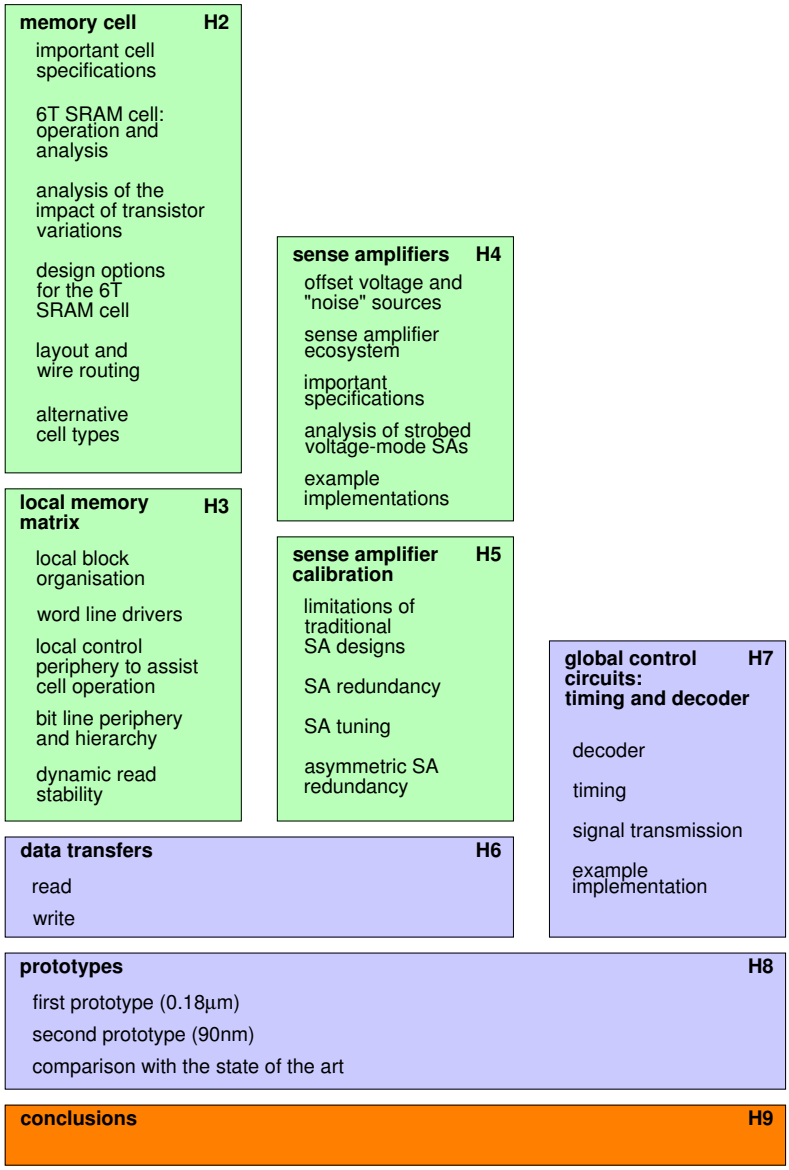


Figure 1.9: Structure of this text.

to SRAM cell layout and local wire routing is included as well. The chapter concludes with an overview of interesting alternative SRAM cell types.

Chapter 3 covers all topics that are related to the local memory matrix. A representative local matrix is sketched, and different implementations for word line drivers are discussed. Then, special attention is paid to practical bias schemes that improve the read stability and write-ability of the 6T SRAM cell. Different implementations of the local bit line periphery are presented and an in-depth treatment of dynamic read stability is provided.

Chapter 4 and 5 address another critical building block of low-power SRAM memories: the global read sense amplifiers. Chapter 4 first introduces the important concept of sense amplifier offset and provides an overview of "noise" sources that negatively affect the sense amplifier operation. Then, the ecosystem in which the SA operates is described and the important specifications for SAs are explained. A practical simulation-based analysis method for strobed voltage mode SAs is then outlined and applied to compare several implementations of strobed voltage-mode SAs. This comparison provides a good impression of the energy versus offset trade-off that must be made with traditional implementations of strobed voltage-mode SAs. This trade-off is a direct consequence of Pelgrom's law for transistor mismatch. Chapter 5 provides a quantitative example of the limitations imposed by this trade-off and analyses and compares different calibration methods that circumvent this limitation: SA redundancy, SA tuning and asymmetric SA redundancy. These techniques can reduce the required differential input signal ΔV_{min} for the SA with a factor of 5 and more for a fixed SA energy consumption.

Chapter 6 discusses the transfer of the data bits between the memory cell and the memory output interface during read and from the memory input interface to the memory cell during write. An overview of bit line structures and signalling types to implement these transfers is provided. For typical implementations of the read operation, both the energy and delay of the data transfer are proportional to ΔV_{min} of the global read sense amplifiers. The bit line write energy in the traditional scheme is 5 to 20 times larger than the bit line read energy. Several techniques to reduce the write energy are discussed. The most interesting approach is to add local low-swing write receivers that amplify the small-swing signal from the global bit lines to a full-swing signal on the local bit lines. Such an approach requires subdivided bit lines.

Chapter 7 outlines the global control circuits, which consist of the circuits for timing control and of the decoder.

Chapter 8 presents the two prototype memories that have been designed, fabricated and measured. They demonstrate the feasibility of the novel techniques as outlined in section 1.4 (except asymmetric SA redundancy). A comparison of their performance with that of other state-of-the-art designs is provided. They clearly outperform the other state-of-the-art designs when low energy consumption per access is required at a reasonable access speed.

Chapter 9 summarises this work and provides some directions for future research.

All examples in this text use the technology data as outlined in appendix C.

Chapter 2

Memory cell

2.1 Introduction

This chapter discusses all aspects of the memory cell itself. Considerations regarding the peripheral circuits that connect to it are avoided as much as possible. These peripheral circuits and their interaction with the memory cell are described in depth in the next chapter.

The memory cell is obviously one of the crucial components of a memory. This chapter first provides a general overview of the important properties of a memory cell in section 2.2. The operation of the SRAM cell is briefly reviewed and the best known metrics to describe cell performance and stability are discussed in section 2.3. In section 2.4, the available methods to obtain a good estimation of performance and yield under transistor variations are outlined. For most of these methods, the quality of the cell metrics that is used is of vital importance. The more smoothly the metrics change with changes in transistor parameters, the easier it is to estimate the distribution of these metrics. In section 2.5, the available design options for the traditional 6T cell are discussed. Section 2.6 briefly treats the layout of SRAM cells in DDSM technologies. Section 2.7 discusses some of the more interesting alternative cell types that have been proposed in literature. Section 2.8 summarizes this chapter.

2.2 Important cell specifications

2.2.1 Area

The area consumed by an SRAM cell is very important. It has two main implications. First, the cell area often contributes significantly to the silicon area and hence to the cost of the total system. Memories consume the bulk of the area in many designs, and the memory area is typically dominated by the memory cells. The ratio of cell area to total memory area is called the area efficiency of a memory. Typical area efficiencies for memories are around 70%. Second, larger cells result in longer wires. This increases the parasitic capacitance and resistance of these wires, which results in a higher energy consumption and delay.

For the relatively small caches which are the main topic of this text, the cost that is associated with the cell area is typically not as important as it is in larger memories. In many applications, it is acceptable to exchange some area for energy savings or speed improvements.

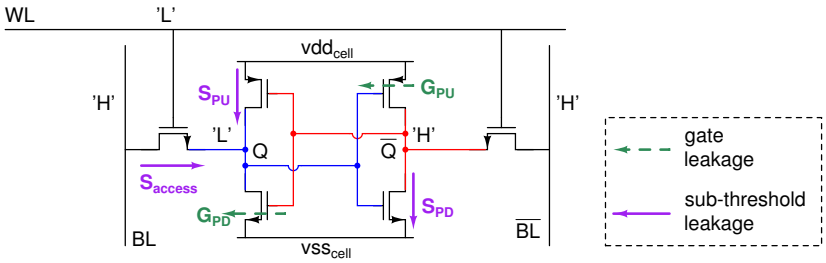


Figure 2.1: The most important leakage paths in a traditional 6T SRAM cell.

In 130nm and 90nm technologies, typical commercial SRAM cell areas are $120 \cdot F^2$ to $140 \cdot F^2$, where F is the critical dimension of the technology node, e.g. $F = 90nm$ for a 90nm technology node [Kha02]. In more recent technologies, the relative size of SRAM cells seems to increase, and the difference in size between different technologies and different designs increases as well. A first reason for this apparent increase in relative size is probably that node names no longer stand in a 1-to-1 relation with technology features, which seems to allow for a rather aggressive naming of nodes. A second, more fundamental reason for the relative size increase is that it becomes harder to satisfy the yield and performance requirements of memory cells without additional transistor upscaling in these advanced technology nodes. Consider for example the 45nm node. Intel has used $170F^2$ SRAM cells [Mis07, Ham09], fabricated with dry patterning. IBM used two different cell designs to satisfy the different requirements for level 1 and level 2 caches [Jos08]. For the level 1 cache, a $230F^2$ high-performance cell was used, while for the level 2 cache a denser $198F^2$ cell was used. The IBM process uses immersion lithography [Nar06]. TSMC presented a cell with an aggressive area of $120F^2$ [Che07] using immersion lithography.

Commercial memory cells do not obey the normal logic design rule checks (DRC). Rather, an analysis is performed to verify whether the specific cell structure can be fabricated reliably. Simulation of the optical lithography process is a critical step in this analysis. Specialized SRAM DRC rules are then be provided to end users. Cell layouts that must satisfy logic DRC rules are significantly larger. Our experience in $90nm$ is that it is hard to get below $240 \cdot F^2$ for 6T cells.

2.2.2 Static power consumption

The static power consumption of SRAM cells is very important for two reasons. Firstly, there are many cells, while only a few of them get activated during each cycle. This increases the importance of static power consumption compared to dynamic power consumption. Secondly, in typical systems, the memory cells cannot be power gated, even during long periods of non-activity. Good system design can resolve this issue for many specific applications, but this is not considered here.

Figure 2.1 shows the most important leakage paths in a typical 6T SRAM cell [Gee05]. A traditional 6T SRAM cell has three transistors with sub-threshold drain-source leakage: one NMOS access transistor, one PMOS pull-up transistor and one NMOS pull-down transistor. Almost all transistors have some gate leakage. However, the gate leakage of the transistors in inversion is much larger than that of the transistors in accumulation [Nii04]. Some authors indicate that junction leakage is important as well in recent technologies [Zha05]. These currents are not indicated on the figure. Whether leakage is dominated by sub-threshold currents, junction currents or gate currents depends on technology options (especially oxide thickness and threshold voltage) and on the cell bias voltages. The difference in sub-threshold leakage between a low-leakage and a high performance technology can be 6 orders of magnitude. In older technologies, sub-threshold leakage was often clearly dominant. In some more recent cases, gate leakage has been reported as the most significant [Nii04, Zha05]. As expected, the recently introduced high-κ metal gate stacks reduce the gate leakage significantly. In [Ham09, Mis07], a reduction in gate leakage with a factor of 25x is obtained by scaling from a 65nm technology with traditional gate stack to a 45nm technology with high-κ metal gate stack. It is clear that the choice of technology has a huge impact on the standby power of the SRAM cell. General rules with respect to leakage and leakage reduction should hence be validated against simulations or measurements in the technology that is actually used. Section 2.3.1.3 will provide a detailed discussion on the leakage currents in the example 90nm technology.

Some memories provide a sleep mode, in which the voltages applied to the cell are altered to reduce its static power consumption. For example, in [Nii04], the SRAM cell leakage is reduced by a factor of 5. An interesting alternative is to keep all cells in a standby mode at all times and to increase Vdd for the accessed cells as part of the memory access [Gee09].

2.2.3 Read delay

In traditional memory designs, the cell read delay is the time between word line activation and the moment at which a sufficient voltage swing ΔV has developed on the bit lines:

$$Delay_{cell} \approx \frac{C_{BL} \cdot \Delta V}{I_{read}}$$

The required magnitude of ΔV depends on the sense amplifiers that are used in the design, as will be discussed in depth in chapters 4, 5 and 6.

Two cell properties influence the read delay in a rather direct way. The cell read current I_{read} is the current that flows through the cell's access transistor during the read access. Typically, an approximate value is obtained from a DC simulation, with the word line voltage at its highest voltage and with the bit lines fixed to their precharge voltage. The larger this current, the less time it takes to develop a sufficient voltage swing on the bit lines. The bit line capacitance per cell $C_{BL,cell}$ is a major contributor to C_{BL}. $C_{BL,cell}$ consists

of the drain capacitance of the access transistor and of the bit line wire and via capacitance related to this cell. The drain capacitance includes both the capacitance of the drain-bulk junction and the drain-gate capacitance. Only one of the cells on a bit line has an asserted word line. The drain-gate capacitance of all other access transistors are biased in the sub-threshold region, which results in a somewhat smaller capacitance value than what would be obtained when all access transistors were enabled.

Increasing the width of the cell access transistor increases the cell read current if the cell remains stable. However, the bit line capacitance increases at the same time, so this will not necessarily result in a large improvement of the cell read delay.

2.2.4 Write delay

The cell write delay is the time between word line activation and the moment that the cell's internal nodes are overwritten. This definition assumes that the bit lines have been forced to their correct value before the word line is asserted. This order of operations is advisable to avoid large short circuit currents.

For memories with a single-cycle access, the cell write delay is almost never a critical factor for the memory delay because the access period is typically dictated by the slower read access.

2.2.5 Read energy consumption

Only a very small part of the dynamic energy consumption of a memory is caused by the capacitances of the accessed cell itself. Typically, the energy consumed to discharge the bit lines is much larger, and this bit line capacitance consists mainly of the $C_{BL,cell}$ of the cells that are not accessed.

For example, consider a memory matrix with fully subdivided WL, and $N = 256$ cells connected to each BL pair. No bit line hierarchy is used in this example. The wire capacitance of WL and BL is ignored for this example. For each accessed bit, there are two contributions to the energy consumption.

Firstly, the WL capacitance is charged. The cell's contribution to this is $C_{WL,cell}$, which consists of the bit line wire and via capacitance related to this cell and the gate capacitance associated with the access transistors. With a traditional 6T memory cell, the energy consumption is around $E_{WL,bit} = 2 \cdot C_{gs,access} \cdot Vdd^2$.

Secondly, for each accessed bit, a full BL is discharged over a voltage ΔV. This BL capacitance is $N \cdot C_{BL,cell}$, which is about $N \cdot C_{drain,access}$, so $E_{BL,bit} = N \cdot C_{drain,access} \cdot Vdd \cdot \Delta V$.

If we assume that the total drain capacitance of a disabled access transistor is comparable to the gate capacitance of an enabled access transistor, the ratio of BL energy to WL energy is $ratio = \frac{E_{BL}}{E_{WL}} \approx \frac{N \cdot \Delta V}{2 \cdot Vdd}$. With $N = 256$ and $\Delta V = 0.1 \cdot Vdd$, the BL energy is 12.5 times larger than the WL energy. Attempts to reduce the dynamic read energy should clearly focus on the BLs rather than

on the WL. A small $C_{BL,cell}$ is crucial. Better sense amplifiers that enable a smaller ΔV can help as well. If a larger energy reduction is needed, subdivided bit lines can be used to further reduce the impact of $C_{BL,cell}$.

2.2.6 Write energy consumption

During a write operation, one of the bit lines needs to be discharged low enough to overwrite the cell contents through the access transistors. The highest BL voltage that is low enough to overwrite the cell data is called the write trip point (WTP), as will be explained in section 2.3. The WTP is typically well below $\frac{Vdd}{2}$, so for write operations, a significantly larger signal swing is needed on the bit lines. In advanced technology nodes, it is probably best to simply discharge the BL to Vss to increase the available cell margin. However, without changes to the SRAM architecture, this results in a huge write energy, up to 10 times larger than the read energy. This write energy is entirely dominated by the BL. Again, a small $C_{BL,cell}$ is mandatory and subdivided bit lines can make a huge difference, especially when combined with local amplifiers, the so-called write receivers, which allow the use of a low-swing signal on the global bit lines.

During write, the internal nodes of the cell can toggle. This consumes some dynamic energy as well. In some cases, short circuit currents are important as well, so they should be correctly analyzed. In cells with a good write-ability, this short circuit energy tends to be rather small.

2.2.7 Bit line load

As mentioned above, the capacitive load that a cell adds to the BL, $C_{BL,cell}$, has a large impact on the read delay, on the read energy and on the write energy.

Therefore, it should be kept as small as possible. Access transistors should have a small width. The cell layout should be shorter in the direction of the BL than in the direction of the WL. The BL should have minimal width, a large spacing to other wires, and the via stack to connect to the cell should probably not be too high, as these vias contribute to the wire capacitance as well. For example, in a 90nm technology, the distance from the top of the active layer to the bottom of metal 3 is about $1400nm$. A typical SRAM layout requires a BL contact once every two cells, and a cell height of $700nm$ is not particularly small. With these numbers, the total via length and the BL metal wire length are comparable. If possible, it is hence better to route the BL in a lower metal layer instead. Some parasitic extraction decks do not take vias into account to calculate parasitic capacitances, so be careful.

2.2.8 Bit line leakage

During the read access, the BL is discharged by the cell read current I_{read} of a single cell. If N cells are connected to the BL pair, and the worst-case data pattern occurs, $N-1$ cells discharge the complementary BL with their leakage currents. In designs where N is large, and the ratio of I_{on} to I_{off} is small,

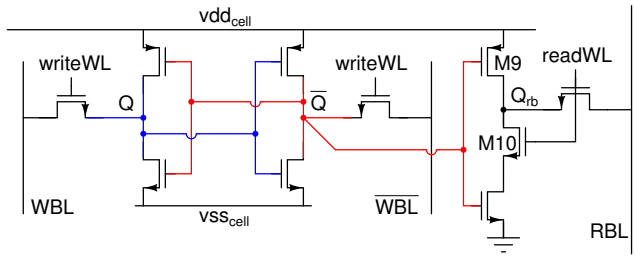


Figure 2.2: 10T SRAM cell that reduces the bit line leakage [Cal07].

the discharge of the complementary BL can be of the same magnitude as the discharge of the actual BL. This can result in incorrect sensing of the data.

This problem occurs when $r = \frac{I_{read}}{N \cdot I_{off}}$ is too small, e.g. if there is 10% margin on the sensing voltage, r smaller than 10 is problematic. For a correct assessment, the variations on I_{on} must certainly be taken into account. It can be important to consider the variations on the sum of the N leakage currents $I_{off,i}$ as well. For large values of N, the impact can be adequately approximated by using the average leakage current. Due to the log-normal distribution of the current distribution, this average value is larger than the nominal value.

This problem seems only relevant in two cases: immature technologies and designs with very low supply voltages, typically sub-threshold memories. Assume a sub-threshold slope of $S = 75\frac{mV}{decade}$. In sub-threshold, $I_{on} = I_0 \cdot 10^{\frac{Vdd - V_{Tn}}{S}}$ and $I_{off} = I_0 \cdot 10^{\frac{0 - V_{Tn}}{S}}$. With a supply voltage of $Vdd = 225mV = S \cdot 3$, the nominal I_{on} over $1 \cdot I_{off}$ ratio is $10^{\frac{Vdd}{S}} = 1000$. Intra-die variations on I_{on} reduce this ratio significantly: with $\sigma_{V_{Tn}} = 40mV$, the 5σ I_{on} is $10^{\frac{Vdd - 5 \cdot \sigma_{V_{Tn}}}{S}} = 2.15$ times I_{off}.

This explains why literature on sub-threshold memories often pays much more attention to the BL leakage than to the total leakage. Several techniques have been suggested to combat this problem.

A rather obvious approach is to reduce the number of cells connected to a BL. [Wan04] pushes this approach to the very limit by replacing the BL with a tree of buffered muxes, each time combining two inputs. Although this results in a very reliable read operation, the area, delay and energy overhead of this approach is impractical [Cal07]. In [Zha08], it was possible to connect 16 memory cells per BL.

In [Cal07], the 10T cell from figure 2.2 was used. It consists of a 6T cell for write and data retention and a separate readbuffer. Only the BL leakage on the read BL is critical. This BL is reduced by adding transistors M9 and M10 to a traditional readbuffer. When Q stores a high value, Q_{rb} is pulled high, so the BL leakage is well controlled. When Q stores a low value, Q_{rb}, there is a stack of 2 disabled NMOS transistors between Q_{rb}, and another disabled NMOS

transistor between the BL and Q_{rb}, so BL leakage is very small.

A very elegant solution was discussed in [Ver08b]. Here, an 8T cell with readbuffer is used. To reduce the bit line leakage, the ground line of the readbuffer is gated on a row-by-row basis. When a word is not accessed, this line is pulled up to Vdd. This greatly reduces the BL leakage. When a word is accessed, all BL currents must be sinked through this ground line and through the NMOS gate transistor. Because the currents are small, the wire resistance is not a problem. The drive current of the NMOS gate transistor is greatly increased by using a higher gate voltage for this transistor. This higher voltage was generated with a local charge pump.

This text will not discuss memory operations at voltages or other operating conditions where BL leakage is problematic, so in this text, the importance of BL leakage is limited to the associated static power consumption.

2.2.9 Cell yield

Immediately after fabrication, some SRAM cells will fail. Some failures will be due to hard physical defects, but those won't be discussed in this text. Other failures will be parametric in nature, caused by variations in the transistor parameters. In this chapter, the focus is on the consequences of uncorrelated intra-die variations. Analysis methods to estimate the impact of these variations are discussed in 2.4. These analyses can be performed in the different process corners, but without more accurate fab data, no global yield can be calculated.

What is a reasonable value for the cell yield Y_{cell}? That is, what cell failure rate $fr_{cell} = 1 - Y_{cell}$ is acceptable? In typical application domains, it is desirable that most of the fabricated chips operate correctly. Although designs targeted at some specialized market segments seem to get away with lower yields, typical yield targets for the entire system-on-chip (Y_{system}) are above 80%. Most systems fail if any one of the memories in the system fails. The memories are not the only blocks that can result in yield loss, and a single block can fail due to many different causes. If there are N blocks, and block i has a yield Y_i, the probability of correct system operation is $Y_{system} = \prod_{i=1}^{N} Y_i$. The available allowed system failure rate should be distributed between the different blocks in a way that minimizes total implementation cost. Consider for example a system with 4 equal memories of 256Kbit and no other blocks with significant chance of failure. If the system yield target is 90%, each memory must have a yield of $\sqrt[4]{90\%} = 97.4\%$.

A plain memory without special recovery features fails if any of its cells fails. Other circuits in the memory, such as the sense amplifiers, can cause failures as well. If only cell failures are considered in this example, the cell yield must be at least $Y_{cell} = \sqrt[256K]{97.4\%} \approx 1 - 10^{-7}$. The allowed cell failure rate is $fr_{cell} = 1 - Y_{cell} = 10^{-7}$. Assume that there is only one failure mechanism (e.g. only read upsets). Assume that the margin M against this failure mechanism follows a Gaussian distribution (e.g. $M = SNM$, the static

noise margin) and assume that cells with $M < 0$ fail. Under these assumptions, the required cell yield is obtained if $fr_{cell} = normcdf(0, \mu_M, \sigma_M)$, or $\mu_M = -norminv(fr_{cell}, 0, 1) \cdot \sigma_M = 5.2\sigma_M$. Notice that the 5.2σ requirement is imposed on the margin M, which is a function of multiple stochastic variables P_i. To meet the requirement on M, the cell will have to operate for deviations of the individual P_i that are somewhat larger than $5.2\sigma_{P_i}$.

In memories that provide error correction or calibration techniques, higher native failure rates can be tolerated. Error correction codes introduce a large area, energy and delay overhead when the word length and memory matrix are small. They will hence not be studied in this text. The most well-known calibration method is the use of redundancy. Faults in memory cells or periphery are detected during a calibration phase and are hidden by redirecting accesses to the affected addresses to redundant words, rows, columns or blocks. Another class of cost-effective calibration techniques changes the operation conditions for weak instances. For example, when a very low sleep voltage is introduced to reduce the leakage power, some cells will fail due to transistor mismatch. When the supply voltage of the few blocks with such weak cells is not reduced during sleep, data retention can be guaranteed while benefiting from a very aggressive reduction in average leakage power. Another example is discussed in [Wan05a]. A memory with two operation modes is used: a high speed mode and a low power mode. While traditional memories must be designed in such a way that virtually all instances meet the delay constraints, the low power mode of this memory is designed in such a way that typical memory instances (e.g. 90%) satisfy the delay constraints. If a memory instance does not meet the delay constraint, it switches to its high speed operation mode. If the system contains many memories and if the delay failures are due to random intra-die variations, most instances will operate in the very aggressive low power mode, and only few need to switch to the high speed mode. This provides an energy reduction compared to the traditional design method. Although some of these calibration techniques are very promising, they will not be considered explicitly in this text. In the remainder of this chapter, a cell failure rate below 10^{-9} for each of the failure mechanisms separately will be considered acceptable to avoid overly complicated calculations.

2.2.10 Reliability

Even if a memory cell operates correctly during the tests after fabrication, it can fail during the lifetime of the device. Transistor characteristics change over time, and the change often depends on the bias conditions [Mar09]. Even if tests are performed on a regular basis, some environmental factors such as temperature and effective supply voltages will fluctuate. Therefore, tests should happen under slightly deteriorated conditions as compared to those during the actual operation, for example at a reduced supply voltage. This text will not discuss how to obtain the optimal deteriorated conditions.

Another important issue that should be considered in the context of reliability

are soft errors caused by ionizing radiation [Kar04], such as alpha particles and neutrons. When such an incident particle travels through silicon, a lot of interactions can occur. It can collide with other particles, which then get accelerated as well. This can cause a cascade of events. From an electrical point of view, the most important effect of the impacting particle is that it generates a lot of free electron-hole pairs in the semiconductor. If this happens in a space charge region of a diode, the charge can get collected, similar to the operation of a solar cell. If this happens at the storage node of a memory cell, the cell state can be flipped. This is called a single event upset (SEU). The rate at which such events occur is called the soft error rate (SER). A single incident particle can also upset multiple adjacent cells. This is called multiple bit upset (MBU). Such adjacent errors can be detrimental for error correction schemes. This problem will not be addressed in this text, but the SER is an important cell specification which should be considered when comparing different memory cells, especially when one considers to reduce the supply voltage and the data retention margins aggressively.

[Kar04] provides a very good introduction to the topic, as well as a good overview of the literature. [Hei08] provides a more recent discussion on the subject.

The SER issue can be addressed to some extent by circuit designers. Increasing the supply voltage has some impact. Increasing the voltage from $0.8V$ to $1.2V$ reduced SER with a factor of 2 [Kar04]. Changing the topology of the circuit has somewhat more effect. In [Haz03], a reduction in latch SER with a factor of 10 was obtained, but at the cost of 40% more area and energy consumption. Changing the technology seems to be more efficient: the SER of circuits in silicon on insulator (SOI) technologies is 5 times smaller than that of normal bulk technologies. If the SOI technology is changed to be hardened against SEU, SER is reduced by a factor of 100 compared to bulk technology [Kar04]. Error correction schemes and codes are probably the most appealing solution to keep the effective SER acceptably low. However, for register files and small scratch-pads, the associated delay and energy overhead of a straightforward implementation is prohibitively large. For those memories, it might be interesting to consider other memory cells with lower SER, such as DRAM cells with large storage capacitors [Iye05], or SRAM cells with dedicated capacitors added to their storage nodes [Sat99]. Another option is to rely on system-level countermeasures [Muk08].

2.3 6T SRAM cell: operation and analysis

This section recapitulates the basic operation principles of the traditional 6T SRAM cell, as shown in figure 2.3. For each operation, the most important analysis methods are described. A few properties of these analysis methods will be highlighted here, but the way these analysis methods can be used to assess behaviour under intra-die variations will be discussed in section 2.4.

In this section, the asymmetric cell that is shown in figure 2.4 will be used

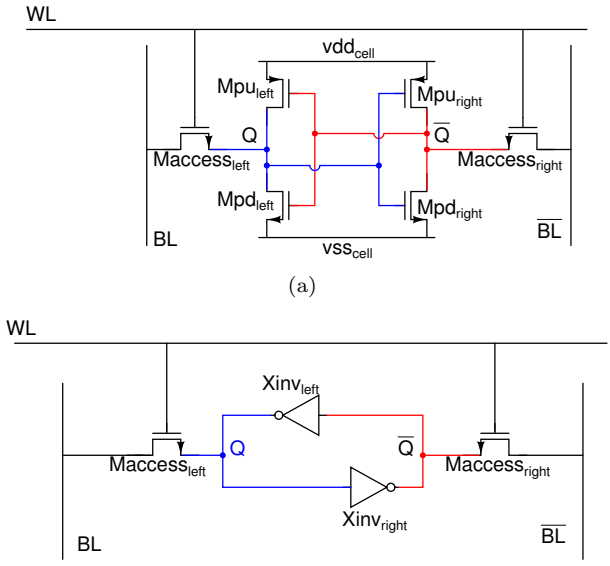


Figure 2.3: Traditional 6T SRAM cell. (a) Transistor schematic. Maccess=access transistor, Mpd=pull-down transistor, Mpu=pull-up transistor. (b) Same cell, different view.

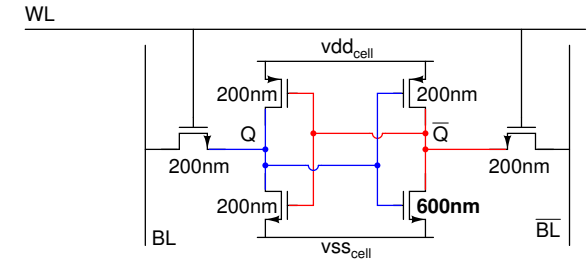


Figure 2.4: The asymmetric 6T cell that is used as running example in section 2.3 and 2.4. All transistors are SPHVT, all transistor lengths are 80nm. Transistor widths are indicated on the figure.

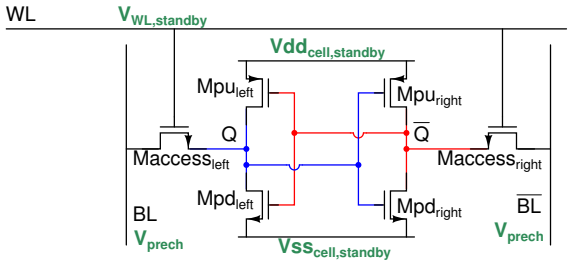


Figure 2.5: Traditional 6T cell during data retention.

as running example. The asymmetry avoids the risk of introducing false assumptions regarding the analysis techniques. In practice, even cell instances of symmetrically designed cells will have some asymmetry between their two sides due to transistor mismatch.

2.3.1 Data retention

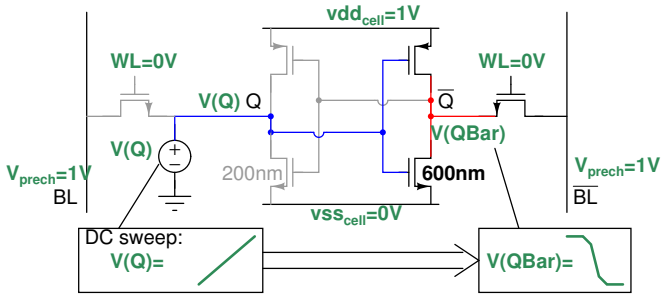
Figure 2.5 shows the cell while it is not being accessed. Traditionally, $V_{WL,standby}$ and $Vss_{cell,standby}$ are both $0V$. The bit lines are fixed at their precharge voltage V_{prech}, which is a relatively high voltage, typically Vdd or $Vdd - V_{Tn}$. With these voltage settings, the access transistors are disabled.

The cross-coupled invertors form a latch, which normally has two possible stable states. At reduced supply voltages or in case of large device variations, one of the stable points can disappear. In this case, the cell does not retain its data correctly. It is also possible that additional, unwanted stable points appear. This is not necessarily a problem, but it does complicate the analysis.

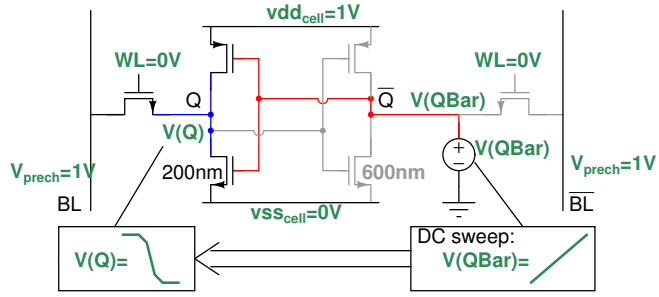
Different failure criteria for cross-coupled latches have been used throughout time. In [Loh83], it was shown that the four most-used worst-case static noise margin definitions are all equivalent. These four are "coincidence of the roots of the flip-flop equation", "unity small-signal closed-loop gain", "Jacobian of the Kirchhoff equations is 0" and "side of the largest square between normal and mirrored voltage transfer characteristic is 0". This last method, the maximum square method, is discussed in [Hil68]. In [See87], analytical equations for this method are derived for full CMOS SRAM cells (6T cells) and for resistor-load NMOS cells (4T+2R). Although the equations are somewhat outdated now, this paper does offer a good introduction to the concept of the maximum square analysis method. [Hau93] provides a good treatment on digital noise margins.

Today, the maximum square method has become the de-facto standard method to analyse cell stability, and the side of the maximum square is simply referred to as the static noise margin (SNM).

Recently, the N-curve based approach has been suggested as an alternative measure of cell stability [Wan05b][Gro06]. This analysis yields not only a static voltage noise margin ($SVNM$) but also a static current noise margin ($SINM$).



(a) V(QBar) when V(Q) is swept



(b) V(Q) when V(QBar) is swept

Figure 2.6: Setup to simulate SNM_{hold}. (a) voltage on Q is swept, voltage on QBar is observed. (b) voltage on QBar is swept, voltage on Q is observed.

This curve can be measured and analysed by automated testers [Wan05b]. It also provides some information about the write operation [Gro06]. The N-curve based approach will not be discussed in this text.

The maximum square method can be applied to both the retention mode and the read mode. The stability is lower during the read access, and SNM normally refers to the margins during the read access. If the margins during data retention mode are discussed, this is typically mentioned explicitly, e.g. SNM_{hold}.

The remainder of this section first discusses the maximum square method as applied to the data retention mode. Then, the data retention voltage (DRV) is introduced and the static power consumption of the cell during data retention is treated.

2.3.1.1 Maximum square method: SNM_{hold}

Definition. The maximum square method is the generally accepted method to calculate the SNM. Figure 2.6 shows the circuit setup that is used in this method. This setup results in the DC transfer curves of both cell halves. The transistors are greyed out at the side where the node voltage is enforced by a voltage source. These transistors have little impact on the resulting transfer curve as their drain-source currents are absorbed by the voltage source. How-

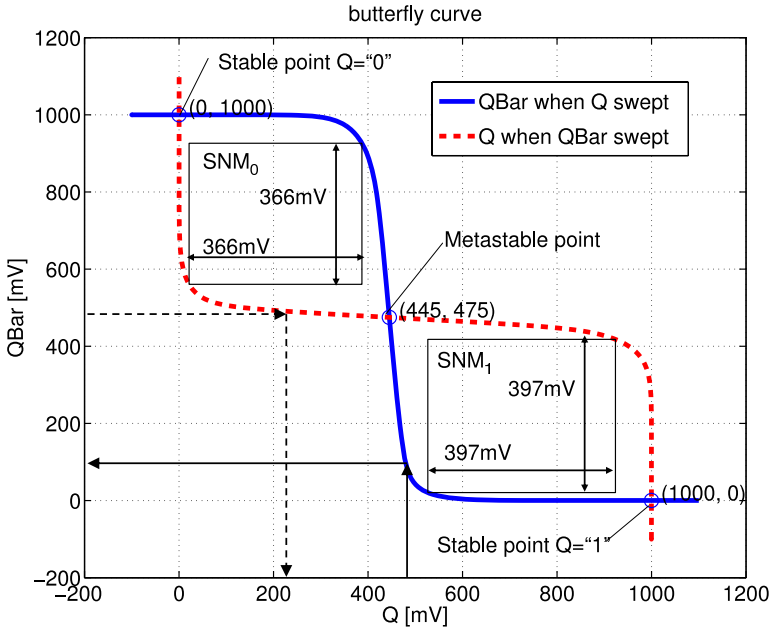


Figure 2.7: Butterfly curves for SNM_{hold}.

ever, they should not be removed from the simulation because their gate leakage can have an impact on the equilibrium voltage at the other node.

The two transfer curves are then plotted on a single graph, as shown in figure 2.7. In this graph, the X-axis is dedicated for the voltage at node Q, while the Y-axis is dedicated to the voltage on node QBar. These dedicated axes are often explained in terms of mirroring the transfer curve around the identity line. That is not necessarily wrong, but keep in mind that the two plotted curves are really different transfer curves: one from Q to QBar, the other from QBar to Q. The resulting graph is often referred to as the butterfly curve. The points where the two transfer curves intersect are possible solutions of the DC analysis of the memory cell. The two outer points are stable points, the inner point is a metastable point. If the state of a memory cell is close to but not exactly in this metastable point, the cell state will move away from the metastable point towards one of the stable points.

The noise margin of a stable point is defined as the side of the largest square that fits between the two transfer curves in the region between that stable point and the metastable point. In practice, it is obtained as follows: rotate the curves over 45 degrees. Find the extreme values of the difference of the two rotated curves. The extreme values correspond to the diagonal of the largest squares [See87]. This approach works for all transfer curves that occur in practice. In this text, the static noise margin of the stable point for the cell state Q = "0" will be designated as SNM_0, while that for the stable point Q = "1" is

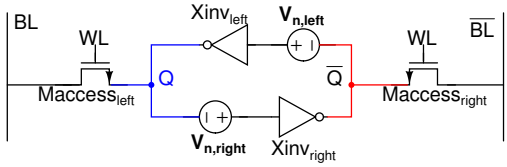


Figure 2.8: A cross-coupled invertor pair with static noise sources $V_{n,left}$ and $V_{n,right}$.

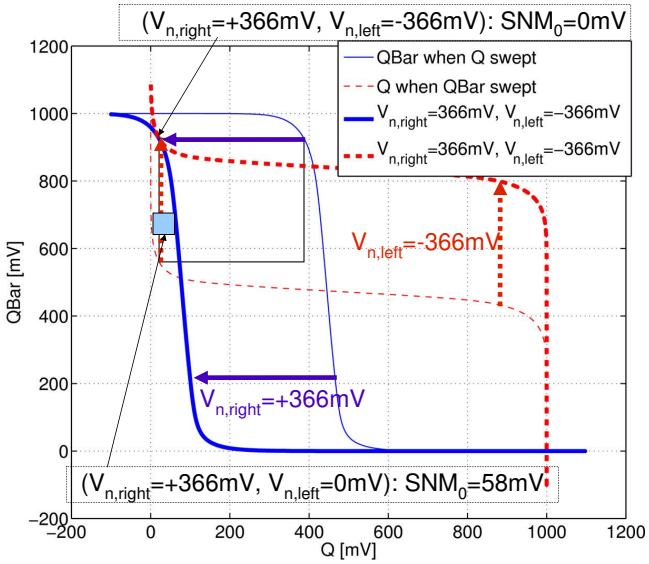


Figure 2.9: When $V_{n,right} = +366mV$ and $V_{n,left} = 0mV$, the transfer curve from Q to QBar (QBar when Q swept) translates to the left. The stable point for $V(Q) = 0$ still exists, but SNM_0 is reduced to $58mV$. The stable point for $Q = 0$ disappears when $V_{n,left} = -366mV$ and $V_{n,right} = +366mV$.

written as SNM_1. The static noise margin of the memory cell is $SNM = min(SNM_0, SNM_1)$.

Impact of transistor mismatch. The SNM was originally intended as a margin against static noise sources such as offsets and mismatches due to processing and due to variations in operating conditions [See87]. These were considered to behave as DC voltage sources in series with the gates of the invertors of the cross-coupled pair, as shown in figure 2.8.

When $V_{n,right}$ is a positive voltage, the input voltage of the invertor at the right cell side (from QBar to Q) will be $V(Q) + V_{n,right}$, a higher voltage than the orginal $V(Q)$. When plotting the transfer curve for this cell half (Q when QBar swept), this results in a translation with value $-V_{n,right}$ along the $V(Q)$ axis, as can be seen in figure 2.9. $V_{n,right}$ does not have a significant impact on the other transfer curve, as it only influences the amount of gate leakage. On the

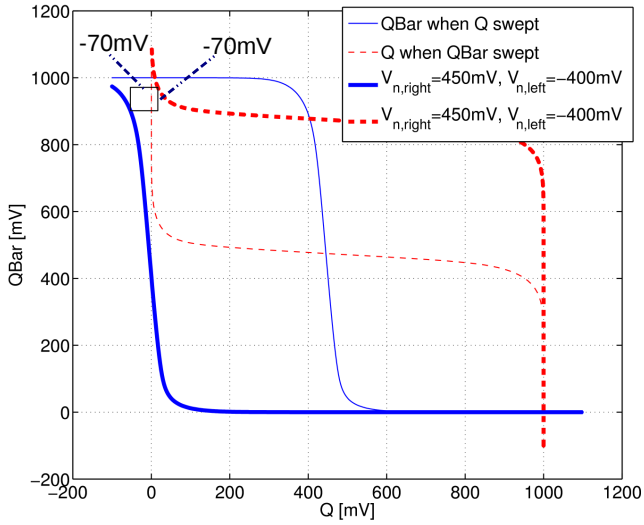


Figure 2.10: The SNM of an unstable cell is negative, with the side of the minimal square between the curves as its magnitude. It indicates the smallest change in $V_{n,left}$ and $V_{n,right}$ that is needed to introduce the second stable point.

other hand, when $V_{n,left}$ is introduced, the transfer curve from QBar to Q will translate with a distance $-V_{n,left}$ along the $V(QBar)$ axis.

In figure 2.9, both noise sources are set to the value of the SNM, each with its worst-case sign. The SNM of a cell with these static noise sources is 0. The metastable point coincides with one of the stable points, and data 3.5ill not be retained. If the noise sources become even larger, the curves will only intersect in the stable point for $Q = 1$. The definition of SNM can be extended to such unstable cells as well, as shown in figure 2.10. Larger negative values of SNM indicate more unstable cells.

Not only the "square" combinations of $V_{n,left}$ and $V_{n,right}$ result in unstable cells, as was discussed in [Hau93]. In our example cell, a trivial combination that leads to failure is $V_{n,right} = +500mV$ and $V_{n,left} = 0mV$. Because the definition of SNM uses squares, both margins on $V_{n,left}$ and $V_{n,right}$ are considered equally necessary and equally valuable. In designs with symmetrical cells, this is exactly the desired behaviour: both curves are equally likely to move due to variations. In cells where there is a large imbalance between the two invertors, measuring squares is actually not the optimal choice. The small invertor suffers much more from stochastic intra-die variations, so its transfer curve is more likely to shift. Even in such cases, there is no real danger in using the normal SNM definition as long as correct statistical analysis is applied. The main disadvantage is reduced insight.

Figure 2.11 shows the SNM_{hold} of each state separately as well as the SNM_{hold} of the cell as a whole. The regions where SNM is small are the most critical

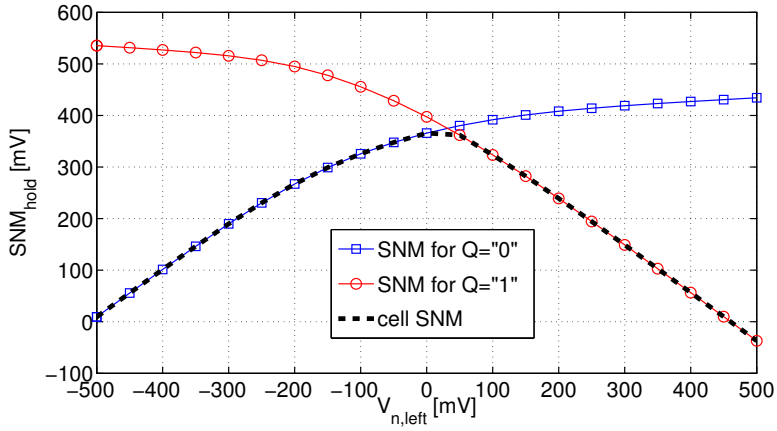


Figure 2.11: The cell SNM_{hold} and the SNM_{hold} of the two states as function of $V_{n,left}$ when $V_{n,right} = 0$.

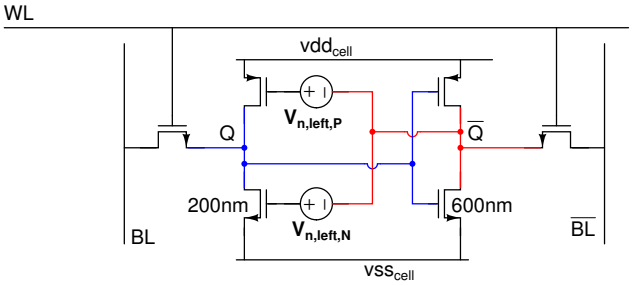


Figure 2.12: Separate static noise sources $V_{n,left,N}$ and $V_{n,left,P}$. $V_{n,right}$ was omitted.

regions. In these regions, the SNM of the individual states varies approximately linearly with $V_{n,left}$. For higher SNM values, there is a strong saturation effect. In the linear region, $|\Delta SNM| = |\Delta V_{n,left}|$. This can be readily understood from figure 2.9.

In figure 2.12, $V_{n,left}$ is further refined into two noise sources: $V_{n,left,N}$ in series with the gate of the NMOS transistor and $V_{n,left,P}$ in series with the gate of the PMOS transistor. This situation is of special interest as it coincides with the way in which the threshold voltage variations of the individual transistors are modeled.

Figure 2.13 shows the impact of these individual noise sources on the butterfly curves. $V_{n,left}$ causes a simple translation, and obviously, the same behaviour is obtained when identical values for $V_{n,left,N}$ and $V_{n,left,P}$ are applied. When the two sources are not equal, the transfer curves change in more complicated ways. In this case, applying $V_{n,left,P} = -366mV$ increases the transistors' gate-source overdrive voltage in the transition region of the transfer curve. The

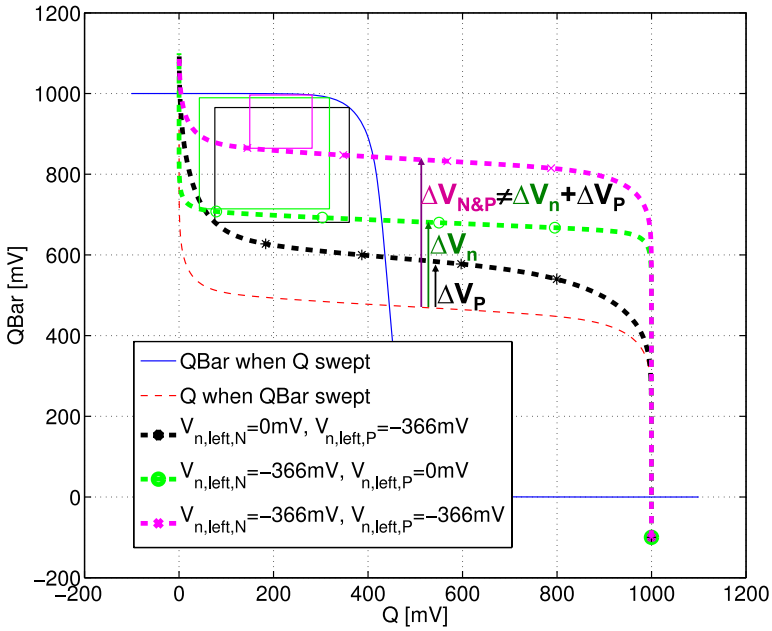


Figure 2.13: The impact on the butterfly curves when $V_{n,left,N}$ and/or $V_{n,left,P}$ are present ($V_{n,right} = 0$).

transistors operate further into strong inversion rather than in sub-threshold or weak inversion, resulting in a more gradual transition region. When $V_{n,left,N} = -366mV$ is applied, the transistors operate deeper in weak inversion or even sub-threshold in the transition region, and the transition becomes more abrupt.

Figure 2.14 indicates how SNM_{hold} varies with the different noise sources. It also shows that the approximation $SNM_{N\&P} \approx SNM_{nom} + (SNM_N - SNM_{nom}) + (SNM_P - SNM_{nom})$, with SNM_{nom} the nominal value of the SNM, is a reasonable approximation, but is far from exact. The fact that the curves in figure 2.14 are not entirely linear and not entirely additive can be problematic when yield estimates are made based on linear response surface models or based on extrapolation of Monte Carlo results.

First-order approximation. An intuitive understanding of the magnitude of SNM values is often useful. For normal cells, the ideal butterfly transfer curves are obtained with idealised invertors with a step transition from Vdd to $0V$ at $V_{trip} = Vdd/2$. In this case, $SNM_1 = SNM_2 = Vdd/2$. In practice, real invertor characteristics deviate from this ideal case in two ways: the transition is not infinitely steep, and V_{trip} is not exactly at $Vdd/2$. Figure 2.15 shows the nominal transfer curves of invertors in a 90nm technology for different supply voltages. Figure 2.15b shows some interesting facts about the SPLVT invertors as compared to SPHVT invertors. At high supply voltages, the transition region is not very steep and with $W_p = W_n$, V_{trip} is much lower than $Vdd/2$, which

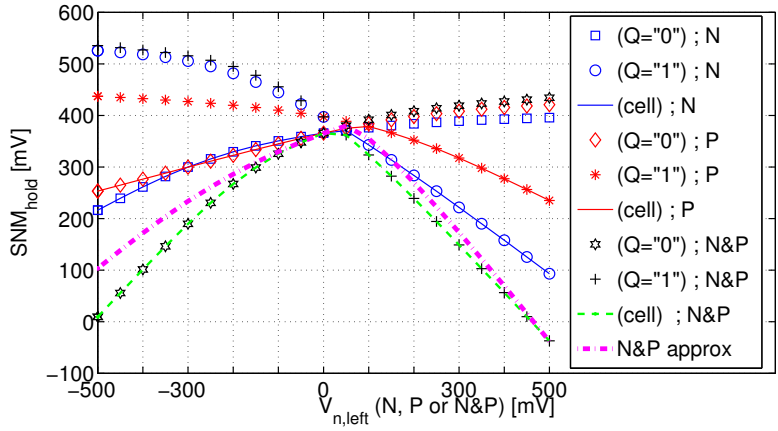


Figure 2.14: The cell SNM_{hold}, $SNM_{hold,0}$ and $SNM_{hold,1}$ as function of $V_{n,left,N}$ (N), $V_{n,left,P}$ (P) and $V_{n,left,N} = V_{n,left,P}$ (N&P). An approximation for SNM_{NandP} is shown as well.

indicates that the PMOS transistor is too weak. Selecting $W_p = 3 \cdot W_n$ shifts the curve up significantly at high supply voltages, but the impact at lower supply voltages is smaller. Figure 2.16 shows $V_{trip} - \frac{Vdd}{2}$ for the different types of invertors. Ideally, this is 0.

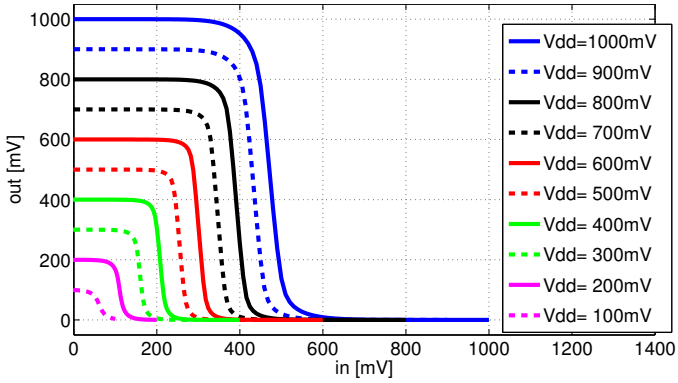
A good approximation for V_{trip} can be obtained for invertors in which both invertors operate in sub-threshold at this voltage [Ver08a]. As V_{trip} should be around $Vdd/2$ and we are especially interested in HVT cells at reduced voltages, this condition is not exceptional. V_{trip} is the input voltage at which $I_p = I_n$ for $V_{out} = Vdd/2$. In sub-threshold, $I_n = c_n W_n exp(\frac{V_{trip} - V_{Tn}}{nkT/q})$ and $I_p = c_p W_p exp(\frac{Vdd - V_{trip} - |V_{Tp}|}{nkT/q}$. These two currents are equal when

$$V_{trip} = \frac{1}{2}\left(Vdd + V_{Tn} - |V_{Tp}| + \frac{nkT}{q} ln(\frac{c_p W_p}{c_n W_n})\right) \tag{2.1}$$

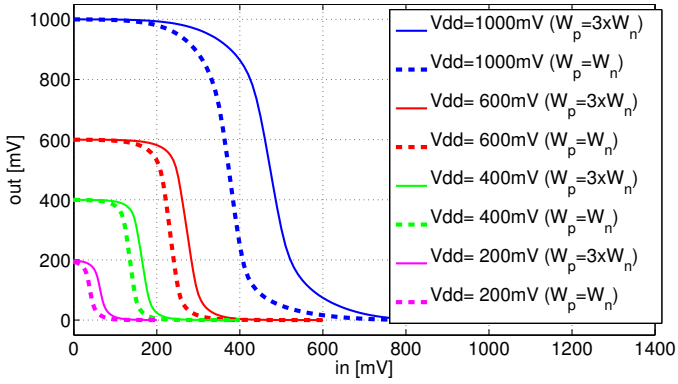
As nkT/q is about $35mV$ at room temperature, changing transistor widths has limited impact on V_{trip}. If W_p is multiplied by 3, V_{trip} increases with $19mV$. In figure 2.16, different increases are observed ($11mV$ and $25mV$). This is due to the small transistor widths. When $W_n = 20\mu m$, the change is indeed $19mV$ at low Vdd. Changes in V_{Tn} and V_{Tp} change V_{trip} with $V_{trip} = \frac{\Delta V_{Tn}}{2} - \frac{\Delta |V_{Tp}|}{2}$.

In case of ideally steep transfer curves, SNM_{hold} is the minimal value of $V_{trip,left}$, $V_{trip,right}$, $Vdd - V_{trip,left}$ and $Vdd - V_{trip,right}$. In this case, one would expect that SNM_{hold} changes with Vdd as $SNM_{hold} \approx C + Vdd/2$. Figure 2.17 shows the behaviour of SNM_{hold} of a nominal cell as function of Vdd. The $C + Vdd/2$ linear approximation through the point at $Vdd = 100mV$ is indicated on the figure as well. It is reasonably accurate for small Vdd.

Notice that it is possible to overcome the $Vdd/2$ limitation for SNM when a



(a) 90nm SPHVT transistors



(b) 90nm SPLVT transistors

Figure 2.15: Transfer curves of invertors at different supply voltages. ($VddBulk = Vdd$)

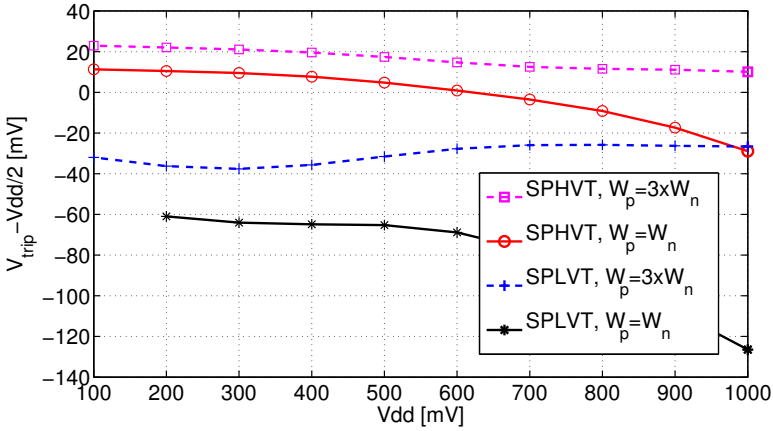


Figure 2.16: Difference between V_{trip} and the ideal trip voltage of $\frac{Vdd}{2}$ for different 90nm invertors as function of Vdd. ($W_n = 200nm$)

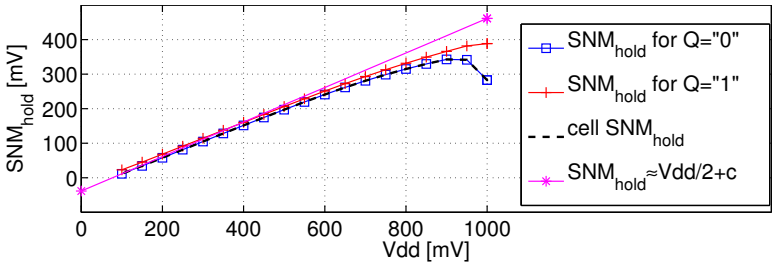


Figure 2.17: SNM_{hold} as function of the supply voltage for the example asymmetric cell from figure 2.4.

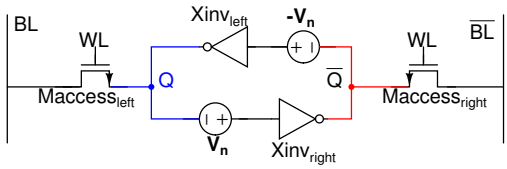


Figure 2.18: Circuit used in the definition of SNM.

cell with a hysteresis effect is employed. Such a cell was proposed in [Kul07] and will be discussed briefly in section 2.7.

A simple simulation approach. Finally, notice that the definition of SNM allows for a very straightforward way to obtain SNM_0 and SNM_1 by means of simulation. SNM_0 is the minimal value of V_n in figure 2.18 which upsets the cell from 0 to 1. This occurs when V_n is positive. SNM_0 can hence be simulated with a slow transient simulation in which V_n sweeps from e.g. $-1V$ to $1V$. The value of V_n at the time of data upset is SNM_0. SNM_1 can be simulated by sweeping V_n in the opposite direction. This approach does not provide the same level of insight as the full analysis of the butterfly curves. However, if only the numerical results are needed, it is a method which is easy to implement and with typical simulators, it is a bit faster as well.

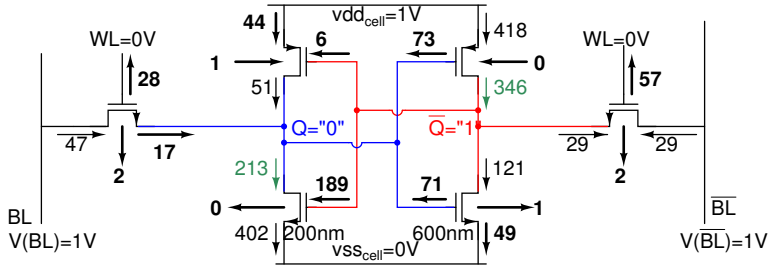
2.3.1.2 Data retention voltage (DRV)

The data retention voltage (DRV) is another interesting metric related to data retention [Wan07b]. The data retention voltage DRV of a cell is the lowest supply voltage Vdd_{cell} at which the given cell retains its data. This is also the supply voltage for which SNM_{hold} becomes 0. The maximal value of the DRV of all cells of a matrix is the minimal supply voltage that can be applied without suffering data loss. Detailed information about obtaining the DRV can be found in [Wan07b].

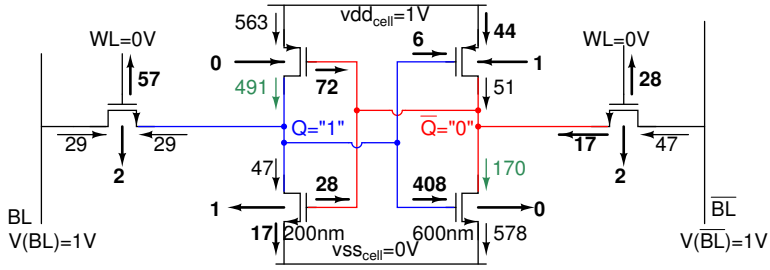
2.3.1.3 Static power consumption (I_{static} and P_{static})

Figure 2.19 shows the leakage currents for our example asymmetric 6T SPHVT cell, at different cell voltages and for both cell states. The bold numbers can be interpreted as the actual undesired current components, while the other currents are either desired on currents that are needed to retain the cell state, or simply the sum of the currents at the other transistor terminals. It is obvious from the figures that gate leakage is rather dominant for this cell at $1V$ supply voltage. When the transistors are replaced with SPLVT transistors, leakage is much larger ($60nA$ at $Vdd = 1V$), and the sub-threshold currents dominate the total leakage. When the temperature is increased from 27°C to 65°C, sub-threshold currents increase with a factor 2.5, while gate currents hardly change.

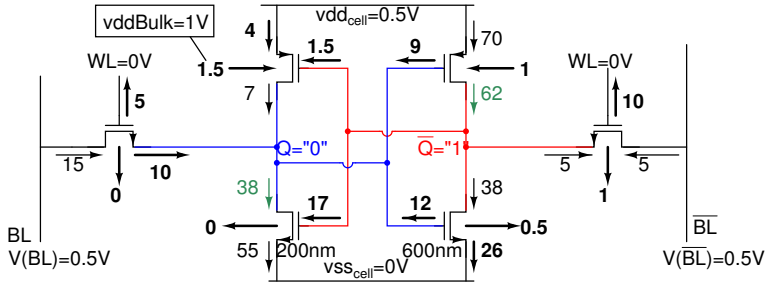
Figure 2.20 shows the behaviour of the leakage current as function of Vdd. At $Vdd = 1V$, the leakage current is $612pA$. At $Vdd = 300mV$, this is reduced to $53pA$, a reduction with a factor 11.5. In case the reduced voltage is pro-



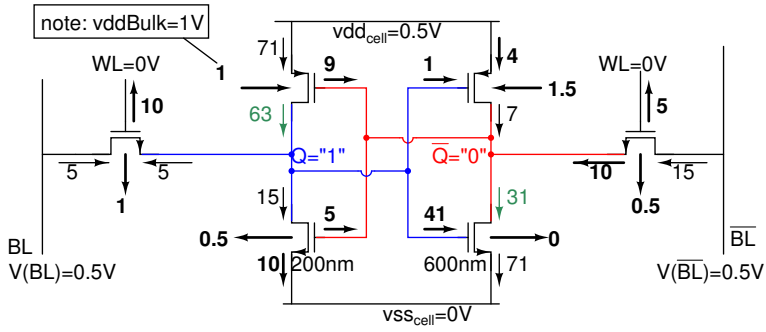
(a) $Vdd_{cell} = 1V$, $VddBulk = 1V$, $Q =$ "0"



(b) $Vdd_{cell} = 1V$, $VddBulk = 1V$, $Q =$ "1"



(c) $Vdd_{cell} = 0.5V$, $VddBulk = 1V$, $Q =$ "0"



(d) $Vdd_{cell} = 0.5V$, $VddBulk = 1V$, $Q =$ "1"

Figure 2.19: The leakage currents at 27°C for different cell voltages and cell states. The numbers next to the arrows are the currents, expressed in pA.

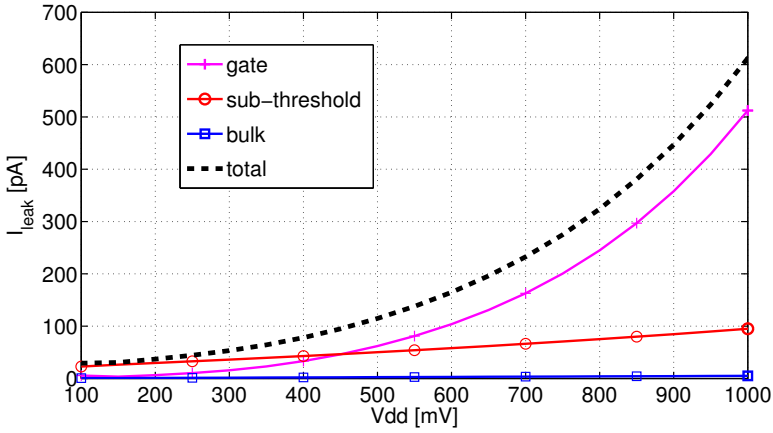


Figure 2.20: Leakage of the example cell as function of Vdd. ($VddBulk = 1V$, $VssBulk = 0V$, 27°C)

vided by a separate power supply or ideal DC-DC convertor, leakage power is $P_{static} = Vdd_{reduced} \cdot I_{static}$. If a series regulator is used, leakage power is $P_{static} = Vdd_{source} \cdot I_{static}$. When the NMOS transistors are reverse biased to $-0.5V$ at $Vdd = 300mV$, leakage is further reduced from $53pA$ to $39pA$.

Gate leakage is dominant when Vdd is above $500mV$, sub-threshold drain-source leakage dominates when Vdd is below $500mV$. Gate leakage is clearly reduced very effectively by lowering the voltage stress over the gate oxides, e.g. by reducing Vdd or increasing Vss. Sub-threshold leakage current is not very sensitive to Vdd and Vss as such. However, reverse body bias has a strong effect. Reverse body bias can be realised by reducing Vdd or increasing Vss while the bulk voltages are kept at their original voltage. In figure 2.19, the PMOS sub-threshold leakage reduces from $44pA$ to $4pA$ when Vdd_{cell} reduces from $1V$ to $0.5V$ while $VddBulk$ stays at $1V$. The sub-threshold leakage of the right NMOS pull-down transistor only reduces from $49pA$ to $26pA$ for the same change in Vdd_{cell}, as this transistor does not receive reverse body bias.

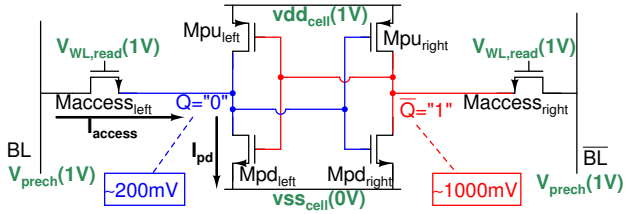


Figure 2.21: A traditional 6T cell during a typical read access.

2.3.2 Read access

Figure 2.21 shows a traditional 6T cell during a typical read access. The bit lines are precharged high, the word line is at Vdd. Assume that the cell state before reading was $Q =$ "0". The pass gate at node Q will draw current from the bit line, and $V(Q)$ will increase a bit. In a DC measurement setup, $V(Q)$ is defined by $I_{access} = I_{pd}$. The PMOS is in cutoff. Typically, the pull-down transistor operates in the linear region, while the access transistor operates in saturation, with node Q as source and with overdrive voltage $V_{WL} - V(Q) - V_{Tn}$. As long as node $V(Q)$ stays sufficiently below V_{trip} of the right invertor, the voltage at $QBar$ does not change appreciably.

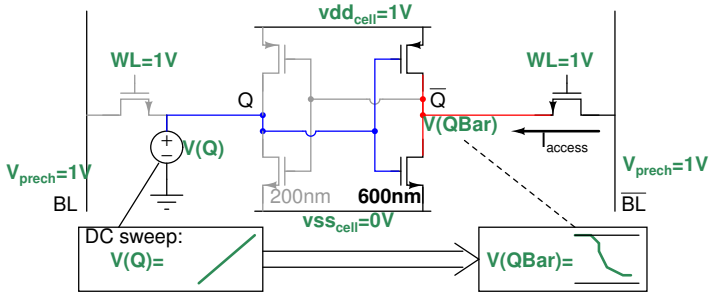
For the read operation, two important requirements must be satisfied: the cell must retain its data and the read operation must be fast enough. The traditional metric that is used to express the margin against read upsets, is the static noise margin SNM as defined by the maximum square method. Sometimes, the operation mode to which the SNM applies, is mentioned explicitly, as in SNM_{read}. This analysis method will be discussed now.

2.3.2.1 Read stability: SNM

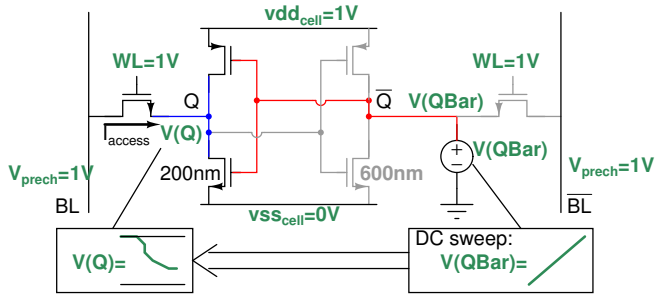
Figure 2.22 shows the circuit setup that is used to calculate the SNM. It is very similar to the setup used for SNM_{hold} as discussed in section 2.3.1.1, but now the word line is asserted. Figure 2.23 shows the impact of different word line voltages on the butterfly curves. The corresponding SNM_{read} is plotted in figure 2.24. The transfer curves of the butterfly curves only change appreciably in the region where the enforced voltage is high. In this case, the invertor must pull the opposite cell node down, but the current I_{access} opposes this. When the enforced voltage is low, the invertor pulls the opposite node high, so the access transistor is in cutoff and $I_{access} \approx 0$. Figure 2.25 shows the voltage on the internal node that stores a "0" as function of the word line voltage.

2.3.2.2 Read delay

The read delay depends on the cell read current I_{read}, on the bit line capacitance C_{BL} and on the BL structure that is used. It is rather straightforward to analyse the nominal delay. It should be noted though that the read current I_{read} is a strongly nonlinear function of typical parameters. Figure 2.26a shows



(a) $V(QBar)$ when $V(Q)$ is swept



(b) $V(QBar)$ when $V(Q)$ is swept

Figure 2.22: The circuit setup to analyse read stability by means of the SNM.

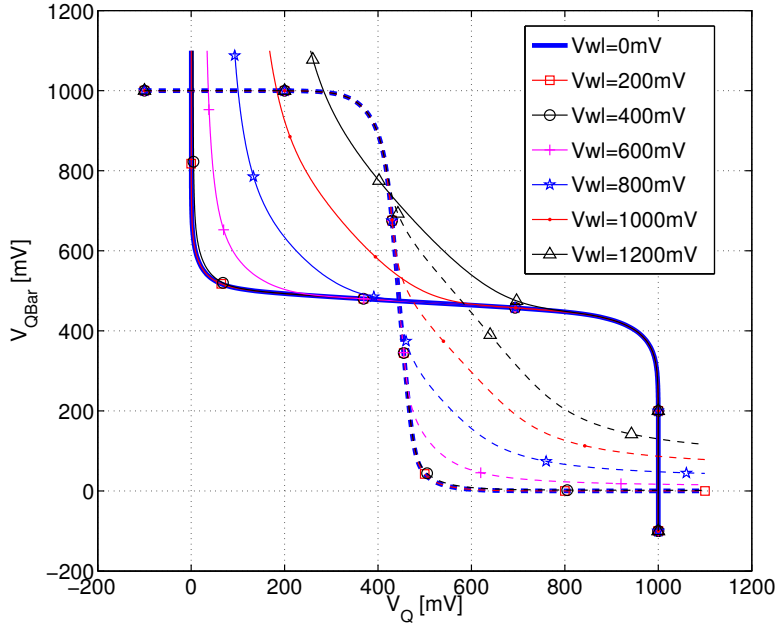


Figure 2.23: Butterfly curves of the asymmetric example cell for different word line voltages V_{WL}. ($Vdd_{cell} = 1V$, $V_{BL} = 1V$)

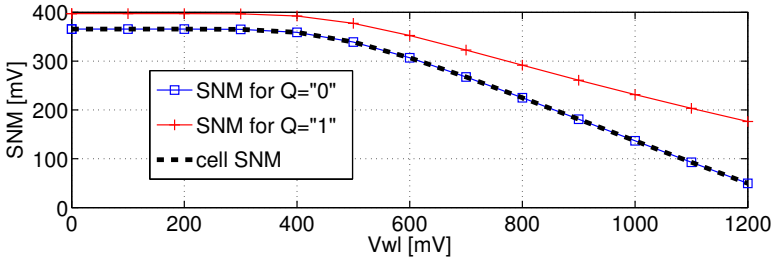


Figure 2.24: SNM_{read} of the asymmetric example cell for different word line voltages V_{WL}. ($Vdd_{cell} = 1V$, $V_{BL} = 1V$)

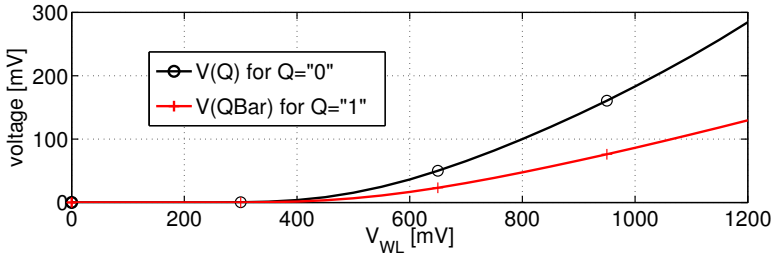
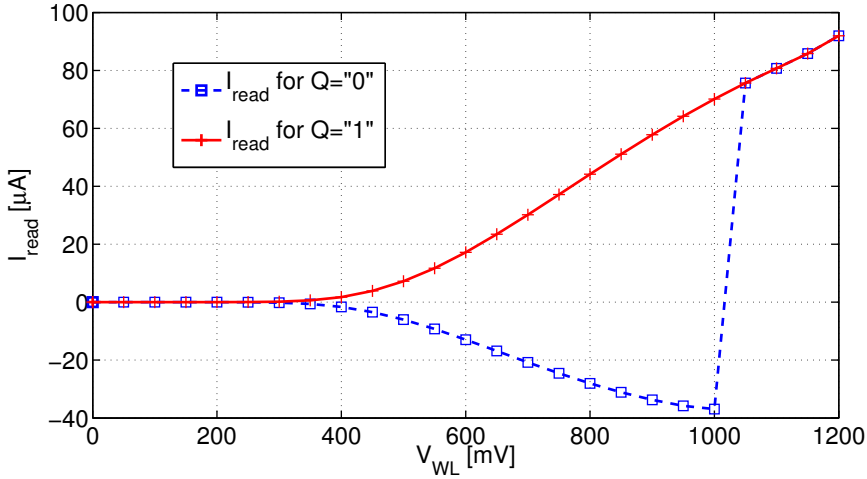


Figure 2.25: The voltage on the internal node that stores a logical 0 of the asymmetric example cell. ($Vdd_{cell} = 1V$, $V_{BL} = 1V$)

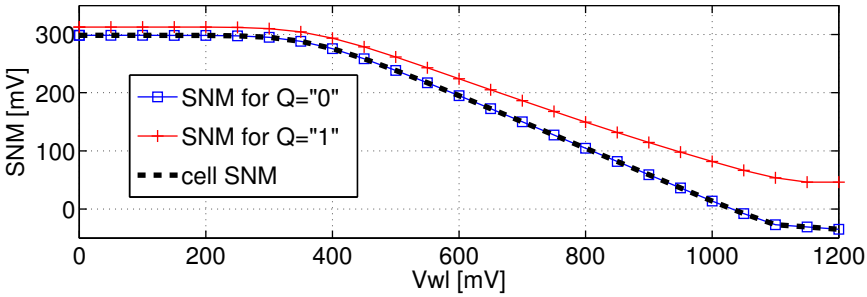
the read current as function of the word line voltage. A reduced cell voltage of $750mV$ was used to clarify the problem. Below $450mV$, cell read current is strongly nonlinear. For memories that operate above $500mV$, the read current is adequately represented by a linear approximation around $V_{WL} = Vdd_{cell} = 750mV$. Such an approximation slightly underestimates the real current. The sudden behaviour at high WL voltage is more problematic: the cell state gets disturbed, and the read current changes accordingly. The I_{read} curve does not give any warning about this. Therefore, any conclusions based on extrapolated values regarding I_{read} should be avoided at all time. Such extrapolation happens for example when a Gaussian distribution is assumed for I_{read}. This sudden point of failure should be captured by using SNM_{read}, for which it is not sudden at all as can be seen in figure 2.26b. This behaviour was adequately described in [Doo08].

2.3.3 Write access

Figure 2.27 shows a traditional 6T cell during a typical write access. In the figure, the initial cell state was $Q =$ "0". The write operation tries to change the cell state to $Q =$ "1" by pulling BL to a high voltage and $BLBar$ to a low voltage. At node Q, the initial situation is identical to that during the read operation. If $QBar$ would stay fixed at its initial voltage, $V(Q)$ would be that of figure 2.25. However, the situation at node $QBar$ is altogether different, as the access transistor now discharges $QBar$. This is counteracted to some extent by



(a) $I_{read} = I_{access,right} - I_{access,left}$



(b) corresponding SNM_{read}

Figure 2.26: I_{read} and SNM_{read} as function of the word line voltage. ($Vdd_{cell} = 750mV$)

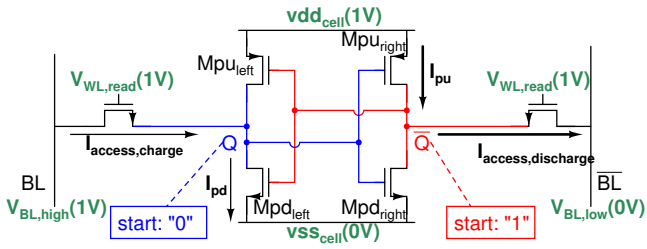


Figure 2.27: A traditional 6T cell during a typical write access.

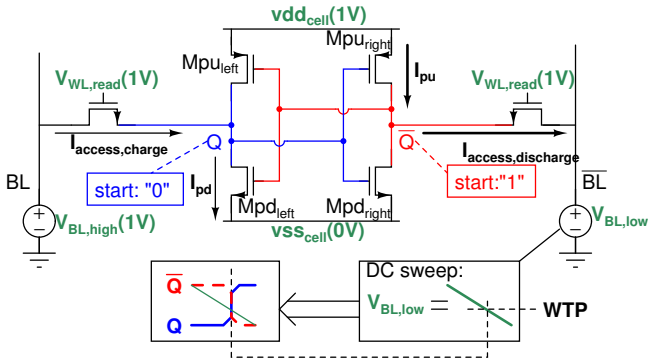
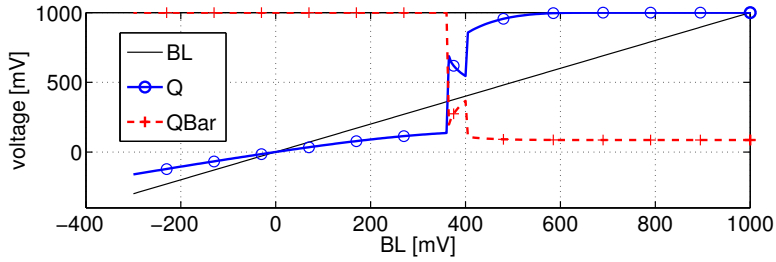


Figure 2.28: Circuit setup to obtain the write trip point WTP

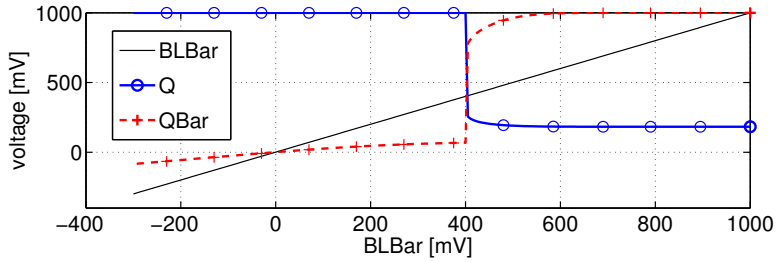
the PMOS current I_{pu}. Notice that I_{pu} is weakened due to the voltage increase on node Q. If the right access transistor is strong enough, node $QBar$ is pulled below the trip voltage of the left invertor. The write operation is completed when the left PMOS transistor pulls node Q up to Vdd.

The traditional metric to describe the write-ability of a cell is the write trip point WTP. It is the highest bit line voltage $V_{BL,low}$ which can overwrite the cell contents. This is typically measured as shown in figure 2.28. Obviously, both writing a "0" and writing a "1" should be considered, and this results in WTP_0 and WTP_1 respectively. Figure 2.29 shows the simulated voltages $V(Q)$ and $V(QBar)$ for both write 0 and write 1. The DC sweep starts from a high bit line voltage, so these plots must be read from right to left. The BL voltage at which $V(Q)$ and $V(QBar)$ cross is called the WTP. The odd behaviour in figure 2.29a just before the WTP deserves some attention. It can be shown that this is a simulation artifact which is caused by the fact that there is no guarantee that DC simulations will return the most likely or closest valid operation point. When the butterfly curve for the cell is plotted, it is obvious that both stable points still exist, but that the simulator jumped to the metastable point anyway. This problem can be avoided by using a slow transient simulation rather than a DC sweep.

Figure 2.30 indicates that for this cell, there is almost a 1-on-1 relation between WL voltage and WTP up to $Vdd_{WL} = 1.2V$. This is expected to some extent, as the main factor that controls $I_{access,discharge}$ is $V_{gs} = V_{WL} - V_{BL,low}$. However, V_{BL} also influences V_{ds} and V_{bs} of this access transistor, while V_{WL} also controls V_{gs} of the other access transistor. Notice that the WTP can be extended smoothly to cells that are not write-able by sweeping $V_{BL,low}$ to negative voltages.



(a) write 0 ($WTP_0 = 360mV$)



(b) write 1 ($WTP_1 = 400mV$)

Figure 2.29: Simulation of the write trip point of the asymmetric example cell.

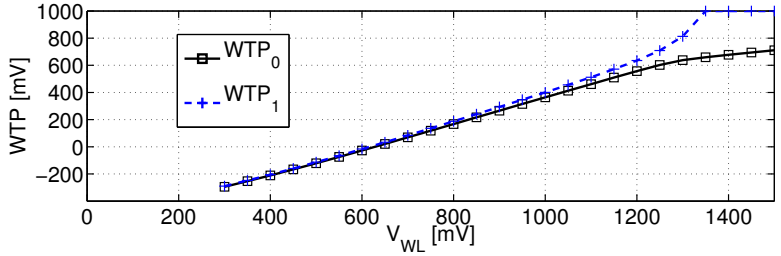


Figure 2.30: WTP as function of word line voltage V_{WL}. ($Vdd_{Cell} = 1V$, $V_{BL,high} = 1V$)

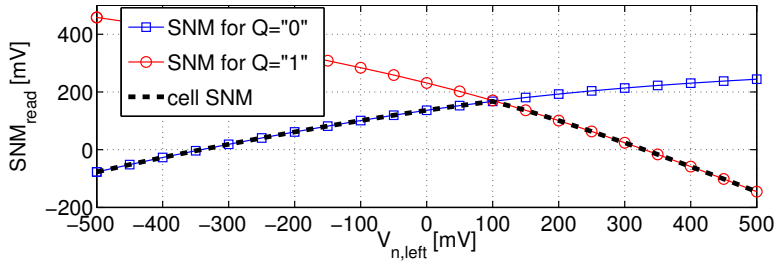


Figure 2.31: SNM_{read} of the asymmetric example cell for different noise voltages $V_{n,left}$, as defined in figure 2.8. ($Vdd_{cell} = 1V$, $V_{BL} = 1V$, $V_{WL} = 1V$)

2.4 Analysis of the impact of transistor variations

This text is mainly concerned with uncorrelated intra-die variations and hence the explanation here will be targeted to these variations. A simplified transistor mismatch model is used in this text in which mismatch is modeled by two parameters, the threshold voltage and β. As there are six transistors in a normal memory cell, analysing the behaviour of a 6T cell under mismatch is a 12-dimensional problem. Different approaches can be used to analyse the impact of transistor variations on the performance of SRAM cells. The most interesting approaches will be discussed now.

2.4.1 Traditional Monte Carlo simulation

2.4.1.1 Method without assumptions

A traditional Monte Carlo simulation takes N samples from the population of cells. In each sample cell, each transistor parameter is assigned a random value which is selected according to its probability distribution. The performance of all N samples are analysed. This results in an approximation of the joint probability distribution of the performance of the population of cells. As we have no special interest at this time in the correlations between the different performance metrics, this text will only discuss the different one-dimensional performance distributions.

The most straightforward and certain approach would be to simulate as much samples as are needed to obtain an approximation of the performance distribution which is accurate enough to support claims about the failure rate without the need to make assumptions with regards to its type of distribution. However, if a cel failure rate fr below 10^{-9} is desired, more than 10^9 simulations would be needed. As each (accurate) cell simulation takes about 1 second on a modern desktop computer, this would take several years to complete.

2.4.1.2 Method with assumptions

A more practical approach makes an informed assumption about the type of the probability distribution of the performance metric. For example, if a Gaussian

distribution can be assumed, it suffices to obtain a good estimate of the mean μ and standard deviation σ of the distribution to know everything about the distribution, and these parameters can be obtained with a few hundred Monte Carlo simulations.

In [Doo08], simulations are presented that show the distribution types of several performance metrics of interest. The read SNM of each stable state individually can be approximated adequately with a Gaussian distribution. This is strongly related with the fact that SNM_{read} varies linearly with the important noise sources, as exemplified by figure 2.31. The distribution of SNM_{read} of the entire cell can then be obtained as the distribution of $min(SNM_{read,0}, SNM_{read,1})$, which is easy as long as it is acceptable to assume that $SNM_{read,0}$ and $SNM_{read,1}$ are independent. A similar approach can be used for the distribution of the write trip point WTP. I_{read} is a slightly more complicated case. A linear approximation underestimates the read currents, but it fails to detect unstable cells. If SNM_{read} is used to detect these cells which suffer from read upsets, a Gaussian approximation for I_{read} underestimates the read current somewhat. This is probably acceptable for cells with a large $Vdd-V_{Tn}$, as the chance that a cell instance lies in the region where the approximation is bad is normally significantly below the desired failure rate. For cells with small or negative $Vdd - V_{Tn}$, the linear approximation will be problematic, as the region where the cell current is severely underestimated will be part of the input parameter space that must yield working cells. The cell delay is typically proportional to $D \sim \frac{1}{I_{read}}$. When small currents do occur in the distribution of I_{read}, the distribution of the delay D will have a heavy tail towards large delay values.

In [Wan07b], it is shown that the distributions of $SNM_{hold,0}$ and $SNM_{hold,1}$ are approximately Gaussian. The distribution of SNM_{hold} can then be calculated from $SNM_{hold} = min(SNM_{hold,0}, SNM_{hold,1})$ under the assumption of independence. They indicate that $SNM_{hold,0/1} \approx k \cdot Vdd + c$, where k is identical for all cells, while c varies from instance to instance. This assumption allows them to calculate the cumulative distribution of the data retention voltage DRV of a single cell by filling out $F_{DRV}(x) = 1 - P[SNM_{Vdd=x} \leq 0]$.

In [Gee09], a monitor is used to measure an approximation of the average SNM_{hold} at the currently applied Vdd. A discussion is provided on how to calculate the minimal monitor SNM_{hold} that is needed to guarantee data retention with sufficient yield. The calculation is based on the distribution of cell SNM_{hold}. The result is used in a feedback loop to reduce Vdd to its lowest safe value.

In [Cal06], the behaviour of SNM_{read} for sub-threshold cells was analysed. A Gaussian distribution was an appropriate approximation for both $SNM_{read,0/1}$ and $SNM_{hold,0/1}$.

For the cell's static power consumption, we are only interested in the average leakage current, which is rather easy to obtain. Notice though that the average leakage will be different from the nominal leakage because the leakage currents do not follow a Gaussian distribution.

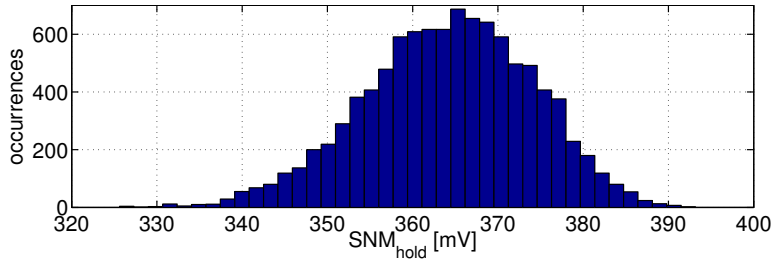


Figure 2.32: Histogram of SNM_{hold} of the asymmetric example cell. ($Vdd_{cell} = 1V$, 10000 Monte Carlo simulations)

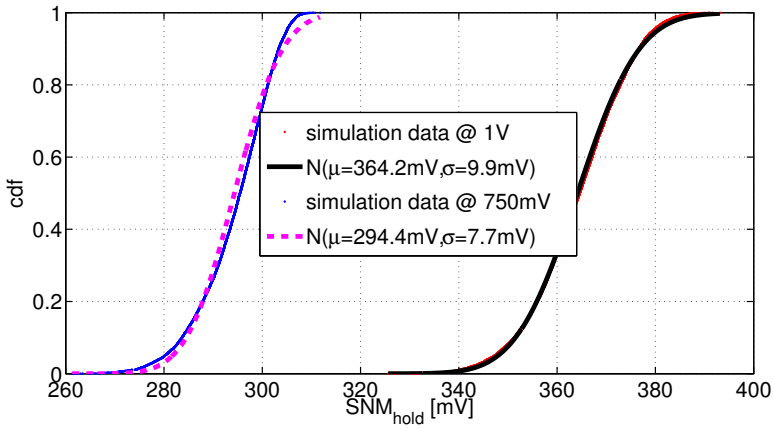


Figure 2.33: Cumulative distribution function of SNM_{hold} of the asymmetric example cell (at $Vdd_{cell} = 1V$ and at $Vdd_{cell} = 750mV$, 10000 Monte Carlo simulations). Both the data points and the Gaussian distribution that is fitted to these data points are shown.

2.4.1.3 Visualisation of results

The raw result of a Monte Carlo simulation is a set of data values sampled from the distribution of the performance metric. The experimental distribution of these data values can be represented in different ways.

Figure 2.32 shows the histogram of SNM_{hold} as obtained by a Monte Carlo run with 10000 evaluated cells. Although this is a widely used and well understood method to visualise experimental distributions, there are good reasons to avoid its use. Firstly, the data in a histogram is divided into discrete bins. Although there are general rules of thumb to select the number of bins to use, the resulting graph will contain less information as the exact performance values are lost. Different artefacts can arise depending on the selected bin size. Secondly, it is hard to get any numerical data from such a histogram, especially about the tails of the distribution.

Figure 2.33 shows the (experimental) cumulative distribution function (CDF)

for the same data. The CDF does not require the selection of a bin size and all available information is displayed. The tail region is still hardly visible though. A plot of $log_{10}(CDF)$ provides a very readable view of the tail with smaller values, as shown in figure 2.34a. A plot of $log_{10}(1 - CDF)$ would highlight the other tail. If both tails are important or if the intention is to verify statistical properties of the dataset, the so-called probability plot is the most suited graphical tool. In such a probability plot, the data are plotted against a theoretical distribution in such a way that the points should form approximately a straight line. Departures from this straight line indicate departures from the specified distribution[NIS06].

Figure 2.34 shows the distributions of the different metrics for the asymmetric example cell. Due to the asymmetry, most of the metrics that are defined as the minimal of two values, such as SNM_{read} and WTP, are dominated by the worst of the two cell halves. This explains why for some metrics the deviations from the Gaussian fit are smaller than what might have been expected. Notice that the leakage currents are plotted on a linear scale in figure 2.34e, as the average value is more important than the tails for this metric.

2.4.2 Faster sample evaluations

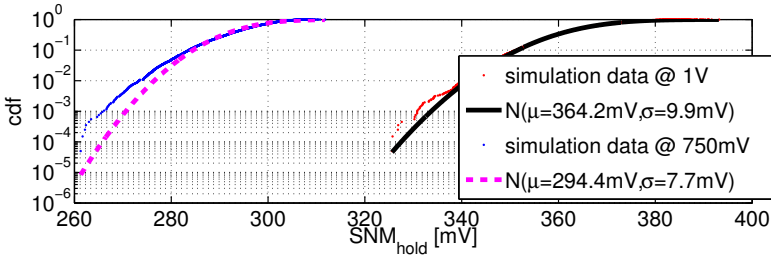
There are two main ways to reduce the time it takes to assess the performance of a sample cell. The first option is to use a faster, probably more dedicated simulator. The second option is to construct a model which approximates the performance metrics as function of the transistor parameters. Such models are often called response surface models and are discussed in 2.4.3.

For a project in which we only had access to lookup tables for transistor currents, not to the compact models, we implemented a simple circuit solver in Java based on these transistor lookup tables, dedicated to the simulation of SNM, WTP, I_{static} and I_{read} for 6T cells. With this rather poor implementation, a speedup with a factor of 50 compared to normal spectre simulations was obtained. Although this result is not too bad, this homemade approach requires a lot of effort and is error prone.

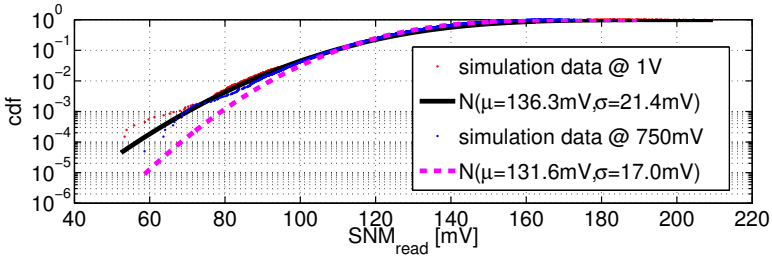
2.4.3 Response surface models

A response surface model is the function that maps the parameter values of the transistors of the cell onto the performance metric that is studied. In the following explanation, $\mathbf{p} = (p_1, p_2, \ldots, p_k)$ is the vector with the parameter values of the transistors of the cell. In the typical 6T cell, $k = 12$. The real response surface from $\mathbf{p}$ to the performance metric that is studied, is called $r(\mathbf{p})$. The metric can be any of the discussed cell metrics such as SNM or I_{read}.

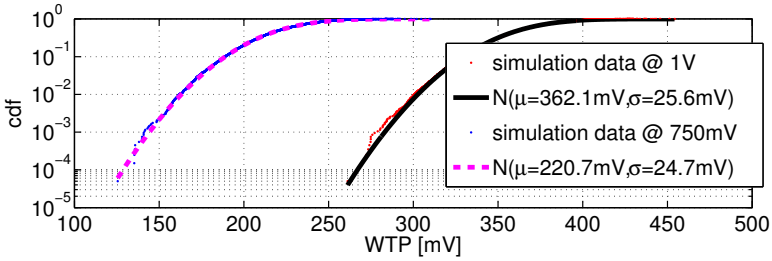
In approaches based on response surface models, an approximation for $r(\mathbf{p})$ is constructed. Once this model is available, the performance of cells is obtained by evaluating the model rather than by simulating the cell itself. The response



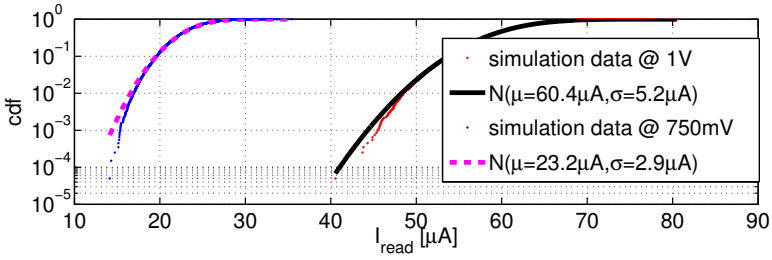
(a)



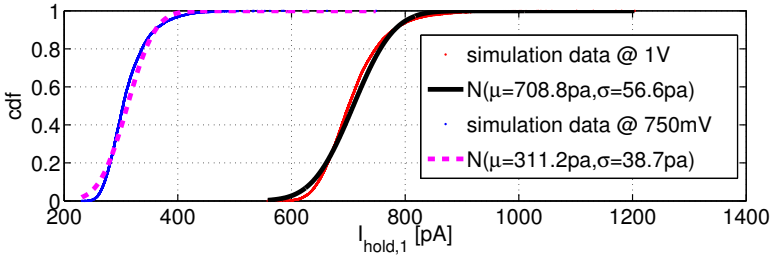
(b)



(c)



(d)



(e)

Figure 2.34: Experimental CDFs for the asymmetric example cell at $Vdd_{Cell} = Vdd_{WL} = V_{BL} = 1V$ and at $Vdd_{Cell} = Vdd_{WL} = V_{BL} = 750mV$.

surface model can then be used to speed up sample evaluation in a Monte Carlo simulation. The model can also be used to assist during the design process.

It is often possible to obtain an approximate analytical model using symbolic circuit analysis. In [See87], closed analytical equations were already provided for SNM_{read}. In [Gro06], approximate formulas for the N-curve around the $SINM$ point were derived. This allows a fast numerical calculation of the $SINM$. In [Cal06], analytical equations for the transfer curves of invertors in the sub-threshold region are employed.

If symbolic circuit analysis is too involved for the considered problem, or if the results should be more accurate than what can be provided by simplified transistor equations, it is still possible to construct an approximate model based on performance values obtained by simulating selected sample cells. The approximate model can be a purely numerical model that uses some interpolation scheme, but it is also possible to fit optimal analytical functions to the response surface [McC05, McC06]. This might be able to provide additional design insight.

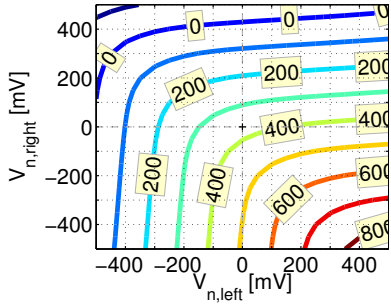
Some practical implementations of sample-based response surface construction deserve special attention. However, before trusting any of these methods, it is necessary to have a good idea of the general shape of the response surfaces. Only SNM_{hold} will be discussed here, and to limit the number of variables, $V_{n,left}$ and $V_{n,right}$ as defined in figure 2.8 will be used instead of the actual mismatch parameters. This is obviously not exact, but it does provide a good overview of the basic type of response surfaces to expect. Figure 2.35 shows the response surface for SNM_{hold} for the asymmetric example cell from figure 2.4 with $Vdd_{cell} = 1V$. The important thing to notice here is that $SNM_{hold,0}$ and SNM_{hold_1} are certainly not linear functions of the mismatch parameters, otherwise all contour lines would be straight lines. This can be understood from figure 2.9. The dominant contribution to the reduction of $SNM_{hold,0}$ is caused by the invertor curve which translates most. Translating the other curve only has a minor impact on the value of $SNM_{hold,0}$. With some rude approximation, one could state that $SNM_{hold,0}$ is defined by $max(-V_{n,left}, V_{n,right})$. Obviously, $SNM_{hold} = min(SNM_{hold,0}, SNM_{hold,1})$ is also not a linear combination of the mismatch parameters.

2.4.3.1 Linear approximation

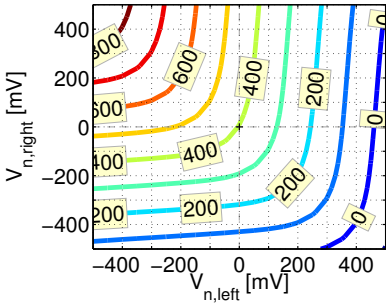
The most simple model that can be constructed is a linear approximation around the nominal point,

$$r_{lin}(\mathbf{x}) = c_0 + c_1 p_1 + c_2 p_2 + \cdots + c_k p_k \tag{2.2}$$

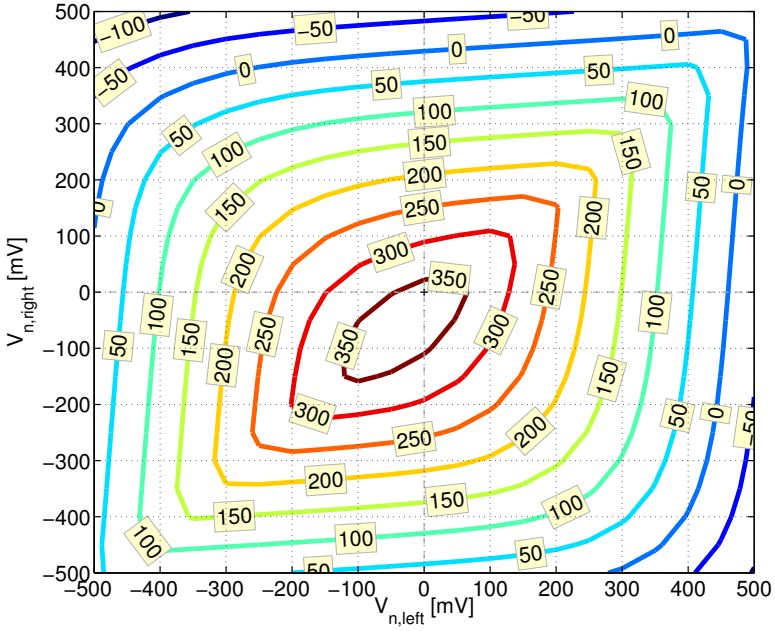
The coefficients c_i can for example be calculated from the simulation results of the nominal point and one point along each parameter axis. If the points along the axes are chosen near to the origin, the coefficients are approximations of the first partial derivative and are often called sensitivities. In this case, this



(a) $SNM_{hold,0}(mV)$



(b) $SNM_{hold,1}(mV)$



(c) $SNM_{hold}(mV)$

Figure 2.35: Example response surface $SNM_{hold} = f(V_{n,left}, V_{n,right})$.

is basically a first order Taylor expansion. Obviously, such a linear approximation is not suited for metrics which include a *min* or *max* operation in their definition, such as $SNM = min(SNM_0, SNM_1)$. Figure 2.35a shows that this problem also exists for $SNM_{hold,0}$ on itself. In [Cal06], similar problems were also observed for $SNM_{read,0}$ in the sub-threshold region.

Calculating the sensitivities or performing 1D sweeps as in figure 2.14 is easy and can provide significant design insights about what is and what is not critical. However, given the shape of the response surfaces of the cell metrics, these sensitivities and 1D sweeps should not be used to calculate parameters of the distribution of the cell metrics, such as $\sigma_{SNM_{hold,0}}$.

2.4.3.2 Grid based approximation

A more accurate solution is obtained when samples are not taken along each axis individually, but in a full grid. In a 2D-problem, samples along each axis individually are points $(x_i, 0)$ and $(0, y_i)$. Samples on the entire grid are all points (x_i, y_j). In a problem with N parameters (dimensions) and K selected points along each axis, there are $N \cdot K$ points along the individual axes, while there are K^N points in the entire grid. A problem with $K = 7$ and $N = 12$ requires $14 \cdot 10^9$ points, which is a bit too much.

A rather simple solution is to eliminate the less important parameters from the analysis. Such an approach is discussed in depth in [Pil08], where it is called a reduced matrix Monte Carlo approach (RMMC), which is somewhat confusing as they actually do not use Monte Carlo simulations. For the read operation, only the threshold voltage variations of 4 devices are considered. For write, the variations of 4 other devices are considered. With $K = 6$ and $N = 4$, only 1296 points must be simulated. It is a workable approach, but a thorough understanding of the circuit operation is needed, and it is easy to overlook important effects.

2.4.3.3 More advanced model generation methods

Much more advanced and efficient algorithms are available to construct approximate models based on the evaluation of points. Good starting points on the subject are [NIS06] and [McC09]. With proper tools and care, it should be possible to obtain an acceptable model in a semi-automatic way. Such automated construction of a response surface is certainly an interesting approach. However, obtaining a reliable response surface in an affordable way is a research field on its own.

2.4.4 More efficient Monte Carlo methods

In the traditional Monte Carlo approach, each transistor parameter of a sample cell is assigned a random value which is selected according to its probability distribution. This obviously means that most of the cells that will be evaluated are rather close to the nominal point. Evaluating the performance of these cells

does not contribute much to our knowledge of the tails of the distribution. In statistics, many different methods have been devised to improve this situation. Some of these methods have already been applied to simulate SRAM yield.

In importance sampling, samples are selected from a different distribution. [Doo08] provides a very clear example. Here, the transistor parameters are selected from normal distributions with an artificially enlarged standard deviation. Without further changes, this obviously results in an incorrect distribution of the cell performance metrics. However, this distortion can be corrected.

For example, consider a specific performance metric, e.g. SNM. In reality, the probability density function of the transistor parameters $\mathbf{p}$ is given by $f(\mathbf{p})$. In a traditional Monte Carlo simulation, we would sample the parameters according to this f. In importance sampling however, the samples are taken according to another probability density function, $g(\mathbf{p})$, which can be chosen to fit our specific needs. N samples $\mathbf{p}_i$ are selected according to g and evaluated, resulting in corresponding samples from the SNM distribution snm_i, with $i = 1 \ldots N$. The actual undistorted cumulative density function of SNM can be approximated as

$$F_{SNM}(snm) = \frac{1}{\sum_{i=1}^{N} w(\mathbf{p}_i)} \sum_{i=1}^{N} (snm_i < snm) \cdot w(\mathbf{p}_i) \tag{2.3}$$

Here, $w(\mathbf{p}_i) = \frac{f(\mathbf{p}_i)}{g(\mathbf{p}_i)}$. The expression $(snm_i < snm)$ evaluates to 1 if it is true and 0 if it is false. Basically, all the samples that resulted in a value snm_i smaller than snm contribute to the value of the cumulative distribution $F_{SNM}(snm)$. To correct for the distorted sampling function, the importance of each sample is assigned a weight.

The approach in [Doo08] seems relatively straightforward to implement. It also seems quite possible to estimate the validity of the obtained results. A more complicated technique was discussed before in [Kan06]. This more complicated technique should be more efficient in problems with many input parameters. However, in the cell problem, there are only 12 significant parameters and the added complexity can probably be avoided.

Another related but different technique is presented in [Sin07, Sin08b]. This specific implementation is called Statistical Blockade. Samples are generated at random, but only those in a region that will likely contribute to the tail region of the performance distribution are simulated. To decide which points should be simulated, a filter is constructed based on a set of initial data points, which can for example be obtained by a normal Monte Carlo simulation. Finally, an appropriate distribution is fitted to the tail region. Obviously, this is a rather involved method to actually implement, and there are a lot of things which might go wrong. The simple importance sampling method seems better suited for analysing the SRAM cell with only 12 parameters.

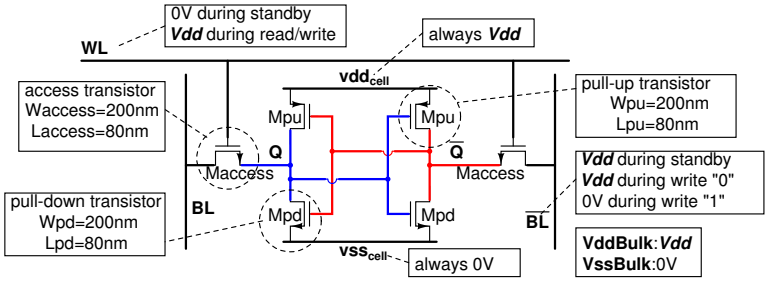


Figure 2.36: The basic symmetrical 6T SRAM cell that is used as baseline in this section. All simulations are performed for $Vdd = 1V$, $750mV$ and $500mV$.

	width $[nm]$	length $[nm]$	V_T $[mV]$	$A_{\Delta V_T}$ $[mV \cdot \mu m]$	σ_{V_T} $[mV]$	$A_{\frac{\Delta\beta}{\beta_0}}$ $[\% \cdot \mu m]$	$\sigma_{\frac{\beta}{\beta_0}}$ $[\%]$
Maccess	200	80	400	4.9	27.3	1.7	9.5
Mpd	200	80	400	4.9	27.3	1.7	9.5
Mpu	200	80	400	3.5	19.6	1.0	5.6

Table 2.1: Basic parameters of the cell from figure 2.36 that is used as baseline in this section.

2.5 Design options for the 6T SRAM cell

This section explores the impact of the available design options for the traditional 6T SRAM cell in a 90nm technology. The cell from figure 2.36 is used as the basic cell in this exploration. The corresponding transistor parameters are provided in table 2.1. With this cell as starting point, a sweep is performed over each available design parameter for three different voltage settings: $1V$, $750mV$ and $500mV$. The available design parameters for this cell are the widths, lengths and threshold voltages of the transistors, the cell supply voltage $VddCell$, the cell ground voltage $VssCell$, the word line voltage $VddWL$, the bit line precharge voltage $V_{BL,prech}$ and the bulk voltages $VddBulk$ and $VssBulk$.

For each evaluated setting, 100 Monte Carlo runs are performed. The corresponding mean and standard deviation will be reported. These numbers are very useful for the purpose of exploration, but they should obviously be treated with proper care. It is useful to first perform an accurate analysis as discussed in section 2.4 for a relevant reference cell to obtain a good impression of the types of underlying distributions. Once a specific setting is selected for implementation, this setting should be subjected again to a thorough analysis.

2.5.1 Accurate distributions for the basic cell

Figure 2.37 shows the experimental CDF of $SNM_{read,0}$, $SNM_{read,1}$ and $SNM_{read,0}$ for the cell of figure 2.36. Notice that the Gaussian approximation seems accept-

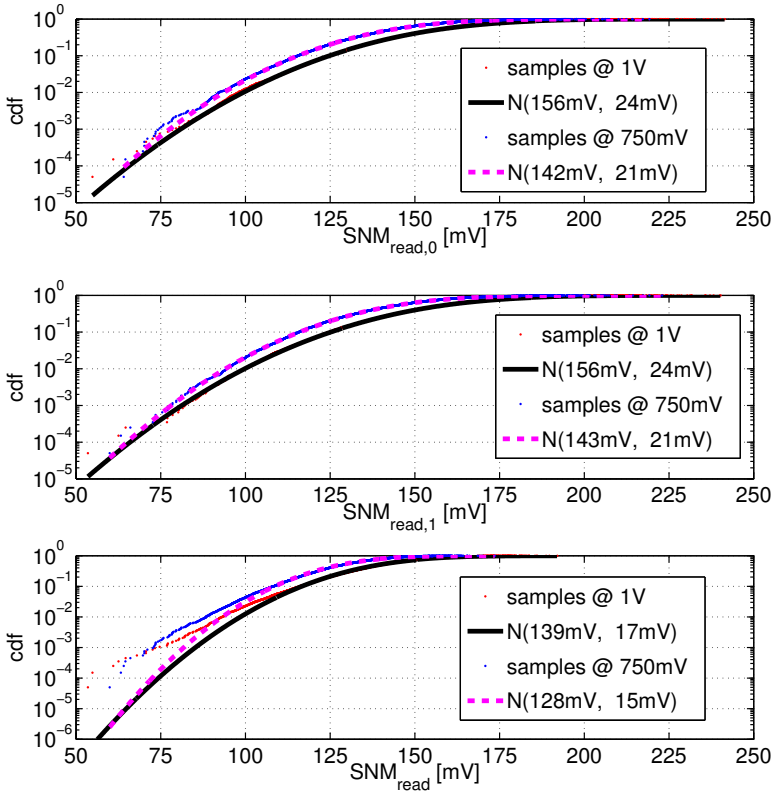


Figure 2.37: Experimental CDF of $SNM_{read,0}$, $SNM_{read,1}$ and SNM_{read} for the cell of figure 2.36.

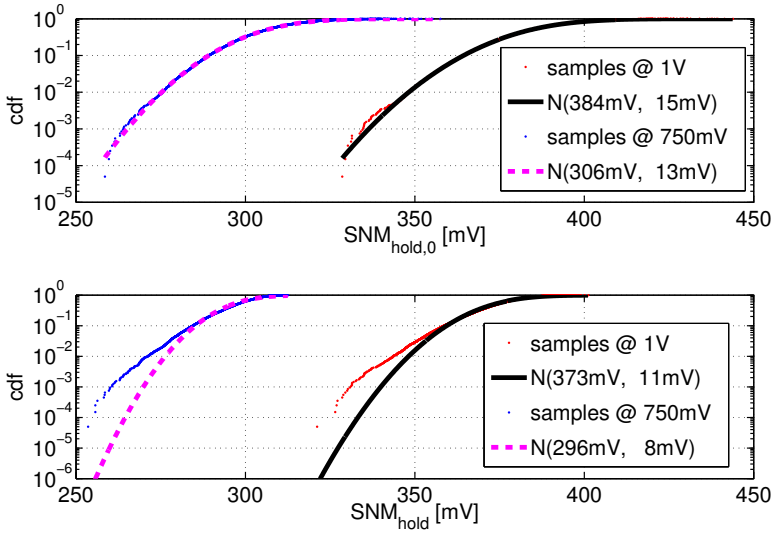


Figure 2.38: Experimental CDF of $SNM_{hold,0}$ and SNM_{hold} for the cell of figure 2.36.

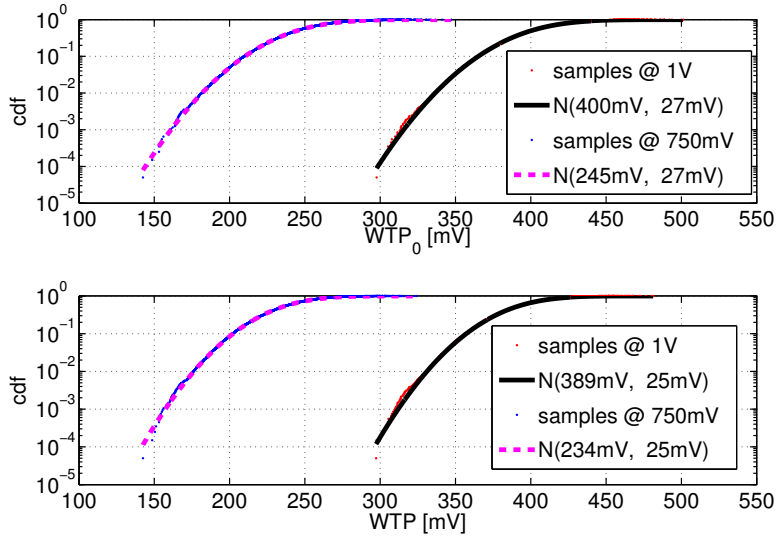


Figure 2.39: Experimental CDF of WTP_0 and WTP for the cell of figure 2.36.

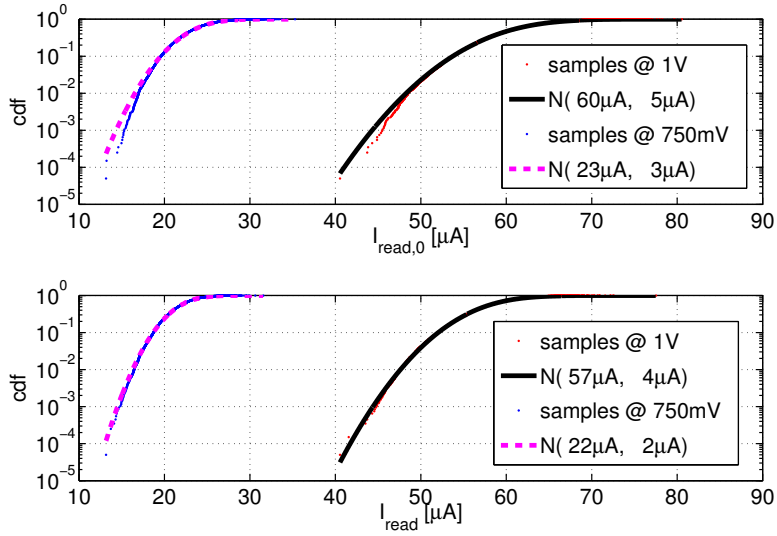


Figure 2.40: Experimental CDF of $I_{read,0}$ and I_{read} for the cell of figure 2.36.

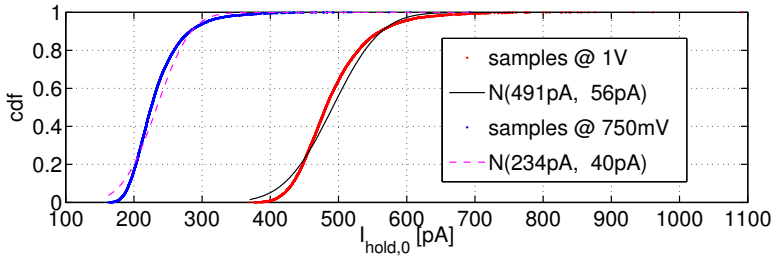


Figure 2.41: Experimental CDF of $I_{hold,0}$ for the cell of figure 2.36.

able for $SNM_{read,0}$ and $SNM_{read,1}$, but certainly not for SNM_{read}. However, in the tail region with low x-values, the graphs indicate that $CDF(x) \approx CDF_0(x) + CDF_1(x)$, so it is fairly simple to make predictions. This is expected, as

$$\begin{aligned} P[SNM_{read} < x] &= P[SNM_{read,0} < x \text{ or } SNM_{read,1} < x] \\ &= P[SNM_{read,0} < x] + P[SNM_{read,1} < x] \\ &\quad - P[SNM_{read,0} < x \text{ and } SNM_{read,1} < x] \\ &\approx P[SNM_{read,0} < x] + P[SNM_{read,1} < x] \end{aligned}$$

The approximation is acceptable if the cell halves are independent enough.

Figure 2.38 shows the CDF for SNM_{hold}. Similar observations as for SNM_{read} can be made. Figure 2.39 shows the CDF for the WTP. All curves seem to be rather well approximated by the Gaussian distribution that is fit to their mean and standard deviation. For $WTP = max(WTP_0, WTP_1)$, this is rather surprising, and there are no indications that this is more than an accidental lucky coincidence. The same remark holds for the CDF of I_{read}, as shown in figure 2.40. Figure 2.41 shows the CDF for the static power consumption $I_{hold,0}$. Obviously, don't approximate this with a Gaussian distribution. A lognormal distribution is slightly better, but still not very accurate.

From the graphs and extracted standard deviations for the two different supply voltages, it is obvious that the shape of the distributions and the standard deviation of SNM_{read}, SNM_{hold} and WTP do not change very much with changes in the supply voltage. This allows us to calculate the lowest average value that can be tolerated for these metrics to stay below a certain cell failure rate, e.g. $fr = 10^{-9}$. The calculations here are based on the data at $Vdd = 750mV$.

To obtain a fr of 10^{-9} due to hold failures, each cell halve must have a failure rate below $fr_0 = 0.5 \cdot fr$. $SNM_{hold,0}$ follows a Gaussian distribution with $\sigma = 13mV$, so the requirement becomes $\mu_{SNM_{hold,0}} > 6.1 \cdot 13mV = 79mV$, which for this case coincides with $\mu_{SNM_{hold}} > 69mV$. At $750mV$, the cell has a mean $SNM_{hold,0}$ of $306mV$, so there is a margin of $306mV - 79mV = 227mV$. It is interesting to calculate the data retention voltage DRV for this cell, as was done in [Wan07b]. Figure 2.42 provides a graphical aid for this analysis. As

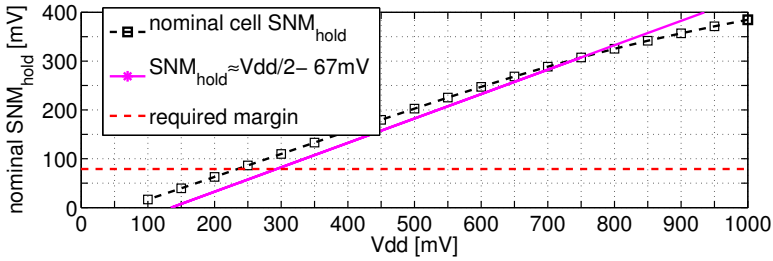


Figure 2.42: The nominal SNM_{hold} for the basic symmetrical 6T SRAM cell as function of $VddCell = VddBulk = Vdd$. This nominal voltage is a good approximation for the mean of $SNM_{hold,0}$, not for the mean of SNM_{hold} itself. The indicated required margin is $6.1\sigma_{SNM_{hold,0}}$.

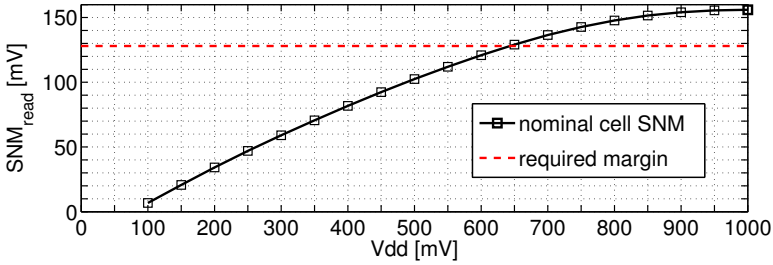


Figure 2.43: The nominal SNM_{read} for the basic symmetrical 6T SRAM cell as function of $VddCell = VddBulk = VddWL = Vdd$. This nominal voltage is a good approximation for the mean of $SNM_{read,0}$, not for the mean of SNM_{read} itself. The indicated required margin is $6.1\sigma_{SNM_{read,0}}$.

σ of $SNM_{hold,0}$ hardly changes with the supply voltage, the required margin of 6.1σ is a constant $79mV$. The available SNM crosses this $79mV$ at $Vdd = 235mV$, so this is the lowest voltage at which the desired cell yield is obtained. An extensive Monte Carlo simulation at $240mV$ supports this estimate. The approximation $SNM \approx Vdd/2 + constant$ that was mentioned before could be used to estimate the data retention voltage as well. If this curve is fitted to the point at $750mV$, the DRV will be overestimated, providing a conservative estimate. When a lower voltage is used for the fit, the approximation can be almost exact.

The minimal nominal $SNM_{read,0}$ can be calculated in a similar fashion, and is $6.1 \cdot 21mV = 128mV$. The read margin of this cell at $VddCell = VddWL = 750mV$ is very small: as the mean value is $142mV$, the margin is only $142mV - 128mV = 14mV$. As figure 2.43 shows, this margin becomes $0mV$ at $Vdd = 644mV$. For WTP_0, the minimal required mean value is $6.1 \cdot 27mV = 165mV$. At $VddCell = VddWL = 750mV$, the mean WTP_0 value is $245mV$, and a margin of $80mV$ is available (in the assumption that the bit line is discharged all the way to $0V$). As figure 2.44 indicates, the margin becomes 0 at $Vdd = 600mV$.

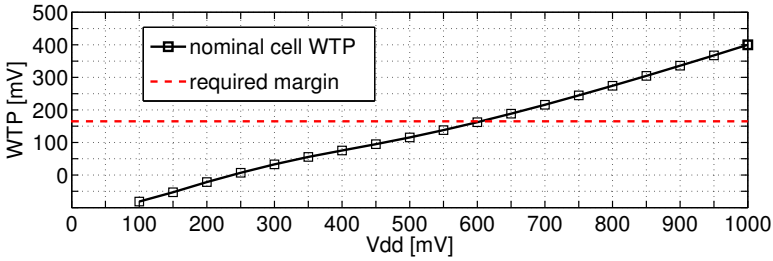


Figure 2.44: The nominal WTP for the basic symmetrical 6T SRAM cell as function of $VddCell = VddBulk = VddWL = Vdd$. This nominal value is a good approximation for the mean of WTP_0, not for the mean of WTP itself. The indicated required margin is $6.1\sigma_{WTP_0}$.

2.5.2 Impact of transistor dimensions and threshold voltages

It is well known that the relative strength of the different transistors has a strong impact on the different cell metrics. At the nominal supply voltage, the transistor operates with large overdrive voltage $V_{gs} - V_T$, and $\frac{W}{L}$ ratio has a large impact on its strength. At lower voltages, the impact of the transistor dimensions on the nominal value of the metrics becomes significantly smaller, as the relative transistor strength is mainly defined by the threshold voltages. As an added benefit, larger transistor dimensions reduce the mismatch in both operation modes.

Figure 2.45 shows the cell metrics as function of the width Wpd of the pull-down transistors. It is well known that increasing the strength of the pull-down transistors increases SNM_{read} while its impact on WTP is relatively small. I_{read} improves quite a bit as well. Although this is not yet visible in the displayed range, increasing Wpd becomes counterproductive beyond a certain point, as this width also lowers V_{trip} of the invertors in the cell, which reduces SNM_{read} as was mentioned in [Cha05a]. At lower supply voltages, the impact of Wpd is significantly lower. To obtain similar results would require a change in threshold voltage of the pull-down transistor. In many technologies, a transistor with lower threshold voltage is available, but the step in threshold voltage is often too large, which would result in excessive leakage currents. The bulk voltage could be used to change the effective threshold voltage, but this affects the access transistor as well. In DDSM technologies, it is often possible to reduce the threshold voltage by increasing the length of the transistors, an effect that is called the reverse short channel effect (RSCE) [Zha08]. In our example 90nm technology, SNM_{read} at $Vdd = 500mV$ increases with only $5mV$ when the length of the pull-down transistor is increased from $80nm$ to $200nm$. The effectiveness of RSCE clearly depends strongly on the process technology.

Figure 2.46 shows the impact of the width Wpu of the pull-up transistors. A cell with a stronger pull-up transistor is harder to write (lower WTP), but has a better read stability (higher SNM_{read}). Figure 2.47 indicates that in this

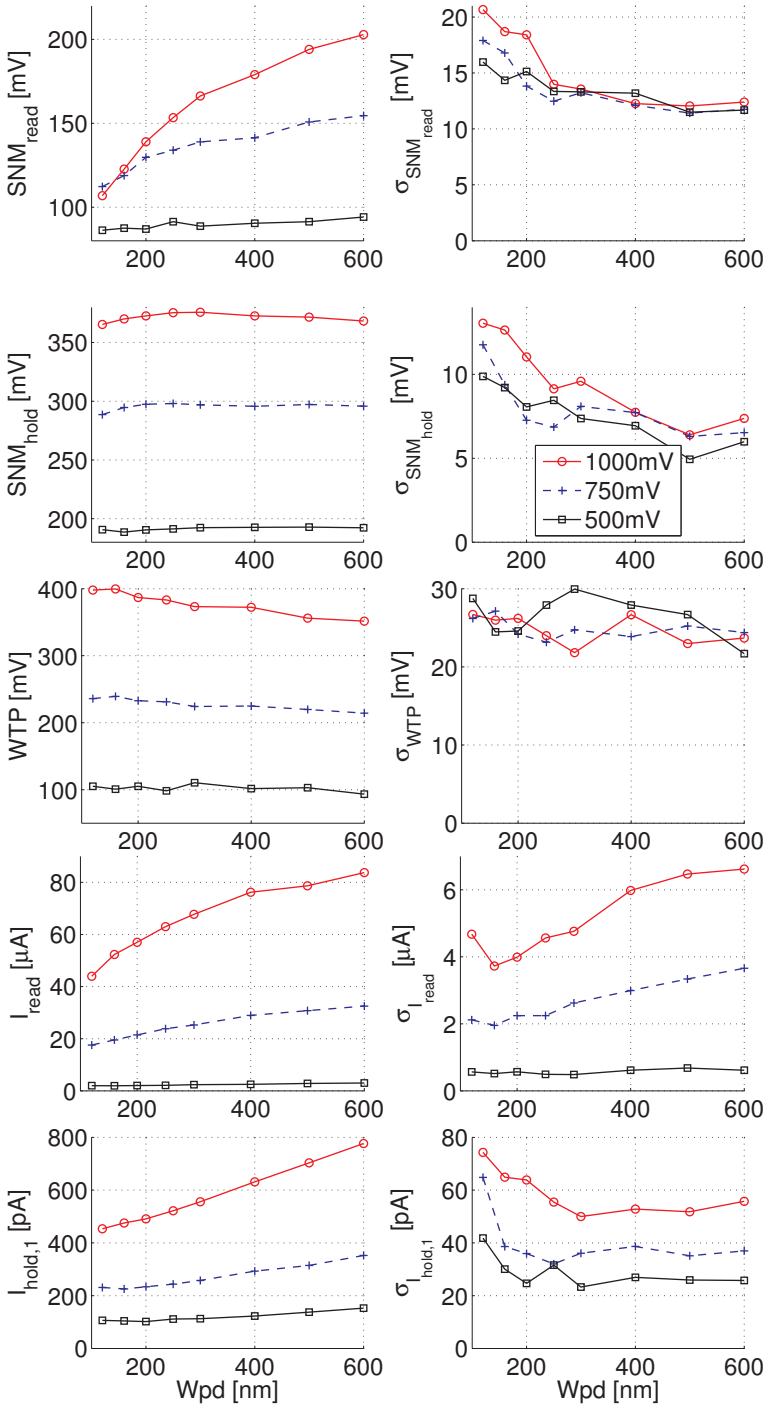


Figure 2.45: Mean and standard deviation for an exploration of the impact of the width Wpd of the pull-down transistors. Data are provided for $VddCell = VddWL = VddBulk = V_{BL,prech} = 1V$, $750mV$ and $500mV$. All other parameters are as in figure 2.36.

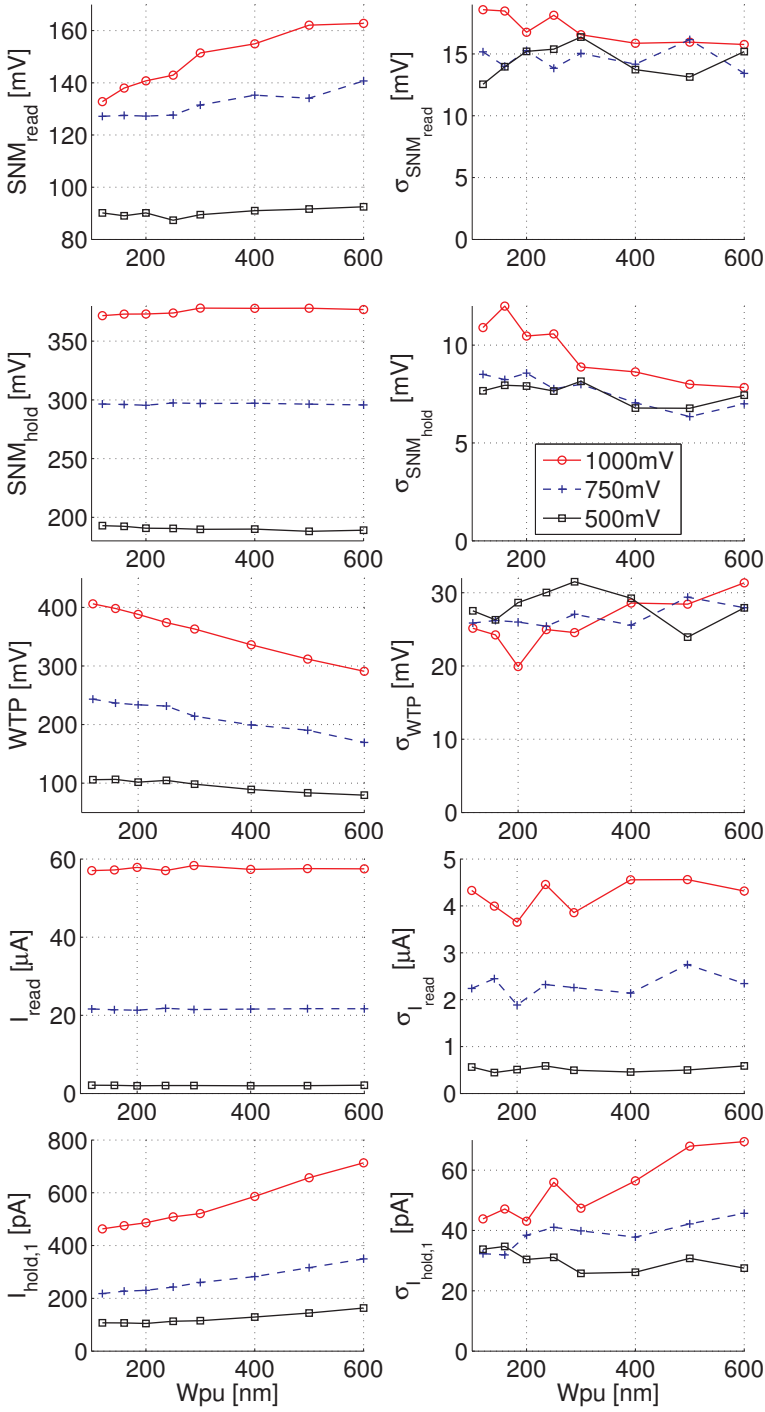


Figure 2.46: Mean and standard deviation for an exploration of the impact of the width Wpu of the pull-up transistors. Data are provided for $VddCell = VddWL = VddBulk = V_{BL,prech} = 1V$, $750mV$ and $500mV$. All other parameters are as in figure 2.36.

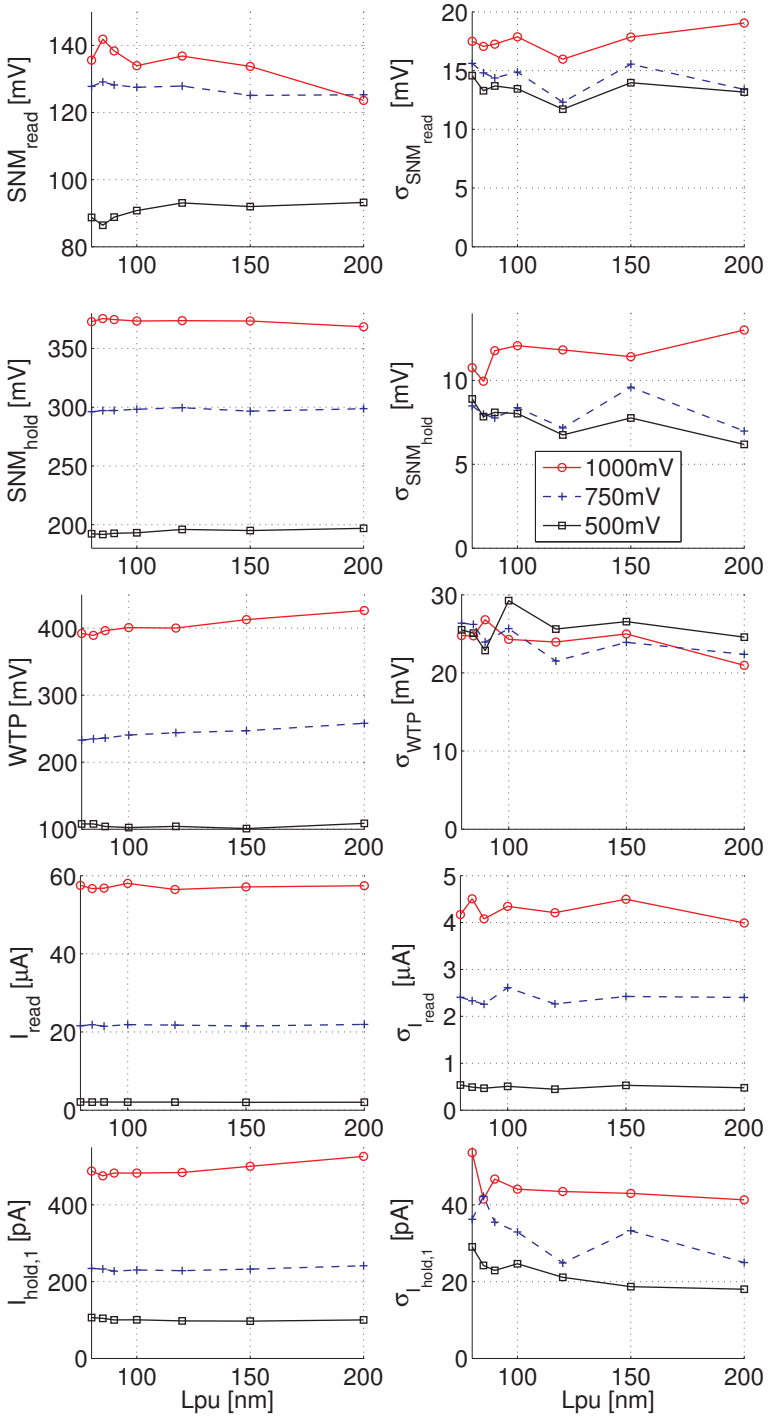


Figure 2.47: Mean and standard deviation for an exploration of the impact of the length *Lpu* of the pull-up transistors. Data are provided for $VddCell = VddWL = VddBulk = V_{BL,prech} = 1V$, $750mV$ and $500mV$. All other parameters are as in figure 2.36.

technology, changing the length L_{pu} of the pull-up transistors has little impact on the WTP.

Figure 2.48 shows the impact of the width $Waccess$ of the access transistor. Keep in mind that increasing this width increases the bit line capacitance as well. In this technology, changing the length of the access transistor has some impact, but significantly less than changing its width.

These graphs illustrate the well-known fact that a 6T cell design must make a trade-off between read stability and write-ability. The only change that makes real sense in this cell with high threshold transistors is a slight increase of Wpd from $200nm$ to $400nm$ to improve read stability and read speed at the cost of write-ability and area.

2.5.3 Impact of cell supplies

Figure 2.49 shows the impact of changing only $VddCell$, while keeping $VddBulk$, $VddWL$ and $V_{BL,prech}$ at a fixed voltage. Larger values of $VddCell$ increase SNM_{read} significantly, while lower values improve the WTP.

The very high values for σ_{WTP} at low values of $VddCell$ are less problematic than they appear at first sight. In this operation region, the access transistor is in the linear region, with its source at the internal cell node. The impact of the bit line voltage is hence small, which results in a large standard deviation. This is however only caused by a heavy tail towards the high values, which means cells that are easily writable.

Figure 2.50 shows the impact of changing $VssCell$ while $VssBulk$ is maintained at $0V$. The magnitude of the impact is similar as that of $VddCell$, but there are significant differences. During data retention, a higher $VssCell$ puts reverse V_{gs} on the access transistors. If sub-threshold leakage is dominant, this reduces their leakage. It also provides reverse back-gate bias to all NMOS transistors. During read, a higher $VssCell$ weakens both the pull-down and the access transistor, so SNM_{read} is less affected while I_{read} is more affected. During write, a higher $VssCell$ increases V_{trip} of the invertor. The strength of the access transistor hardly changes, as it is the bit line terminal that acts as the source.

2.5.4 Impact of bulk voltages

The bulk voltage provides a practical way to change the effective threshold voltage of the transistors in the cell. In typical layouts, all NMOS transistors in the cell share the same bulk voltage $VssBulk$, while all PMOS transistors share the same bulk voltage $VddBulk$.

Figure 2.51 discusses the impact of $VssBulk$, which affects the threshold voltage of both the access transistor and the pull-down transistor. Negative values for $VssBulk$ put a reverse bias over the source-bulk and drain-bulk junction diodes, so junction leakage is not a problem. A negative $VssBulk$ increases the NMOS threshold voltage. This weakens both the access transistor and

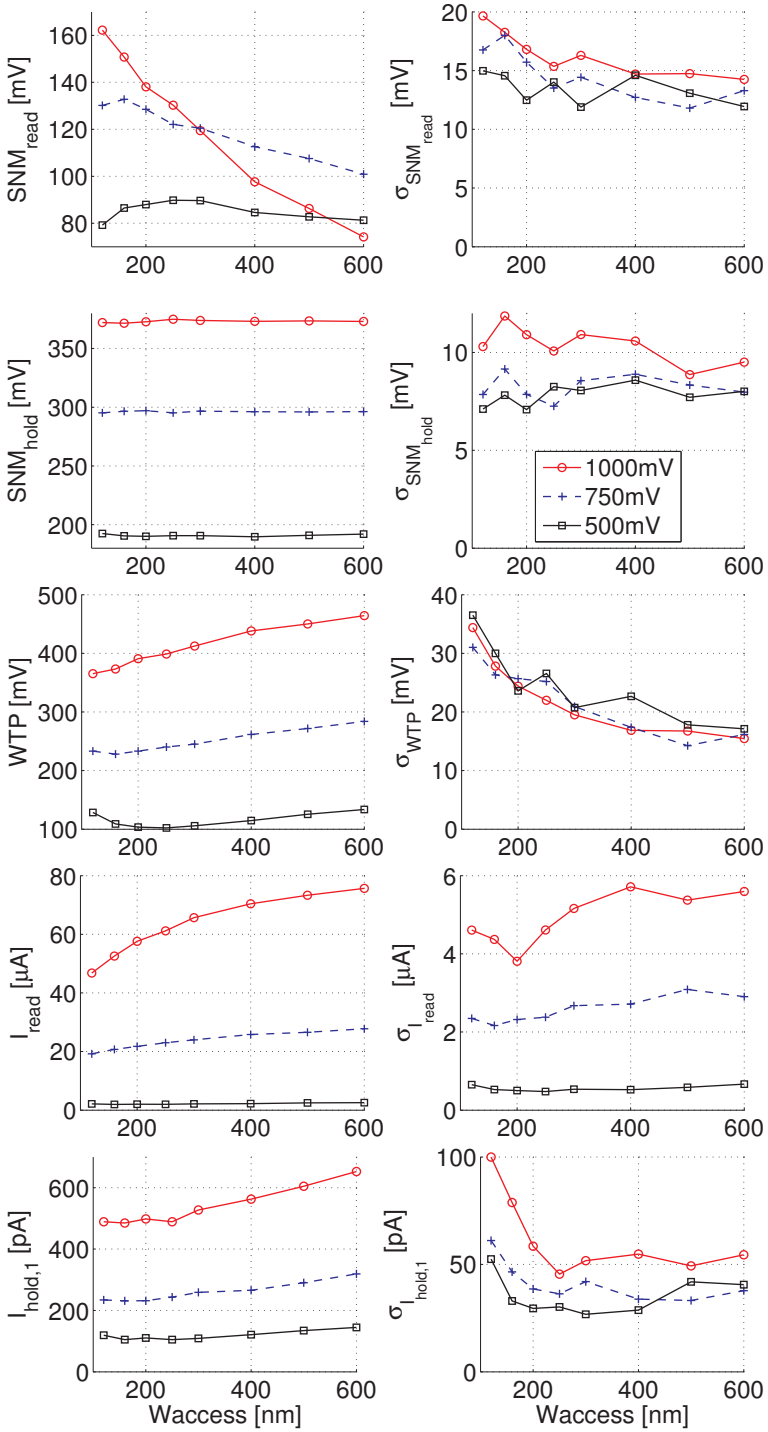


Figure 2.48: Mean and standard deviation for an exploration of the impact of the width $Waccess$ of the access transistors. Data are provided for $VddCell = VddWL = VddBulk = V_{BL,prech} = 1V$, $750mV$ and $500mV$. All other parameters are as in figure 2.36.

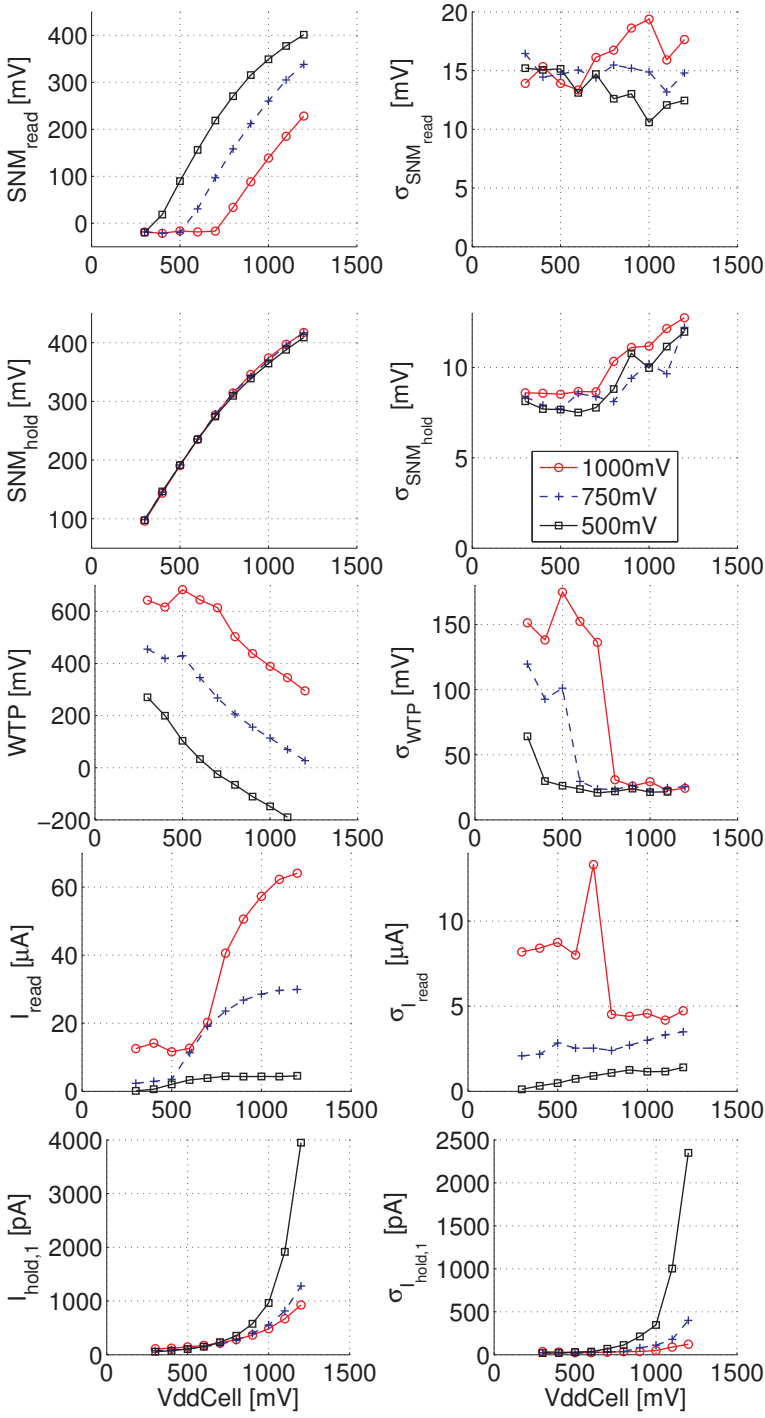


Figure 2.49: Mean and standard deviation for an exploration of the impact of the cell supply voltage $VddCell$. Data are provided for $VddWL = VddBulk = V_{BL,prech} = 1V$, $750mV$ and $500mV$. All other parameters are as in figure 2.36.

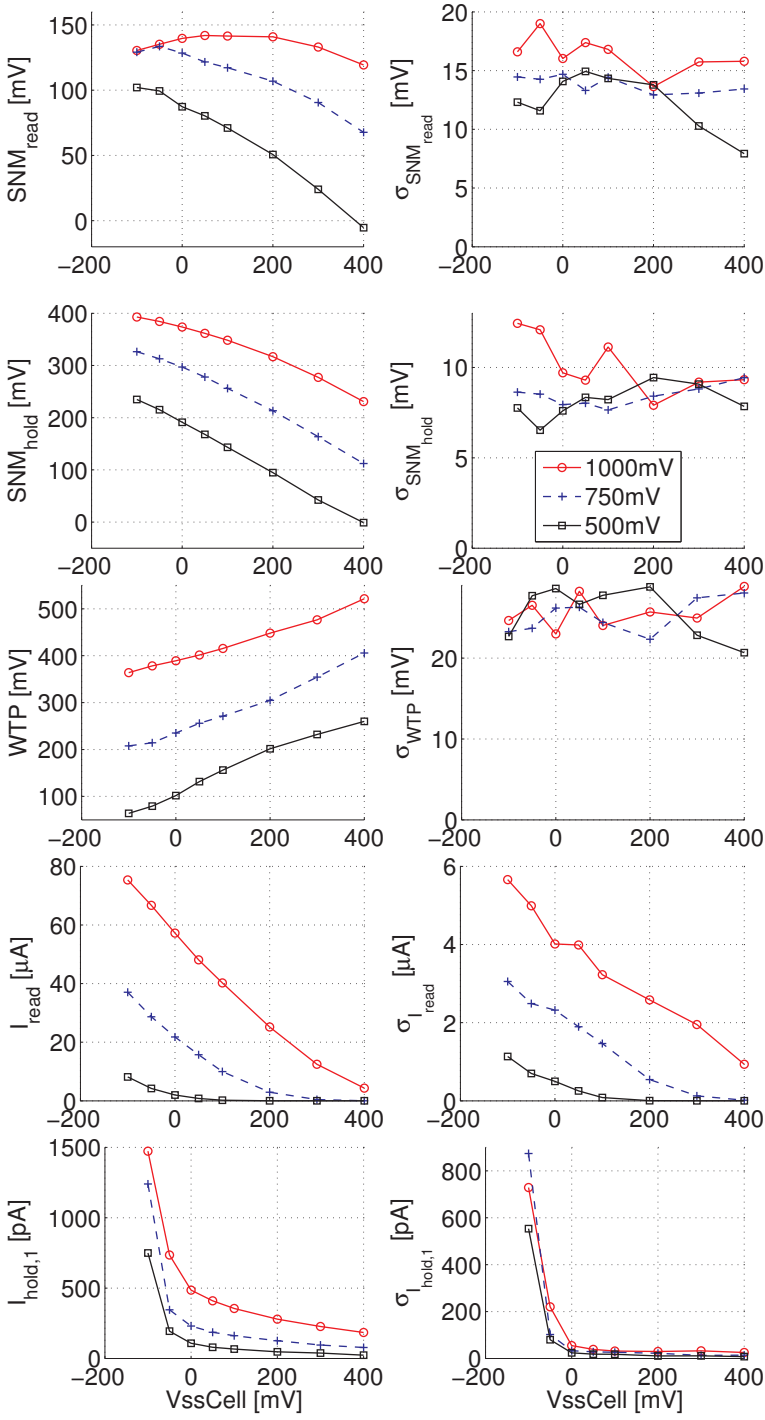


Figure 2.50: Mean and standard deviation for an exploration of the impact of the cell ground voltage $VssCell$. Data are provided for $VddCell = VddWL = VddBulk = V_{BL,prech} = 1V$, $750mV$ and $500mV$. All other parameters are as in figure 2.36.

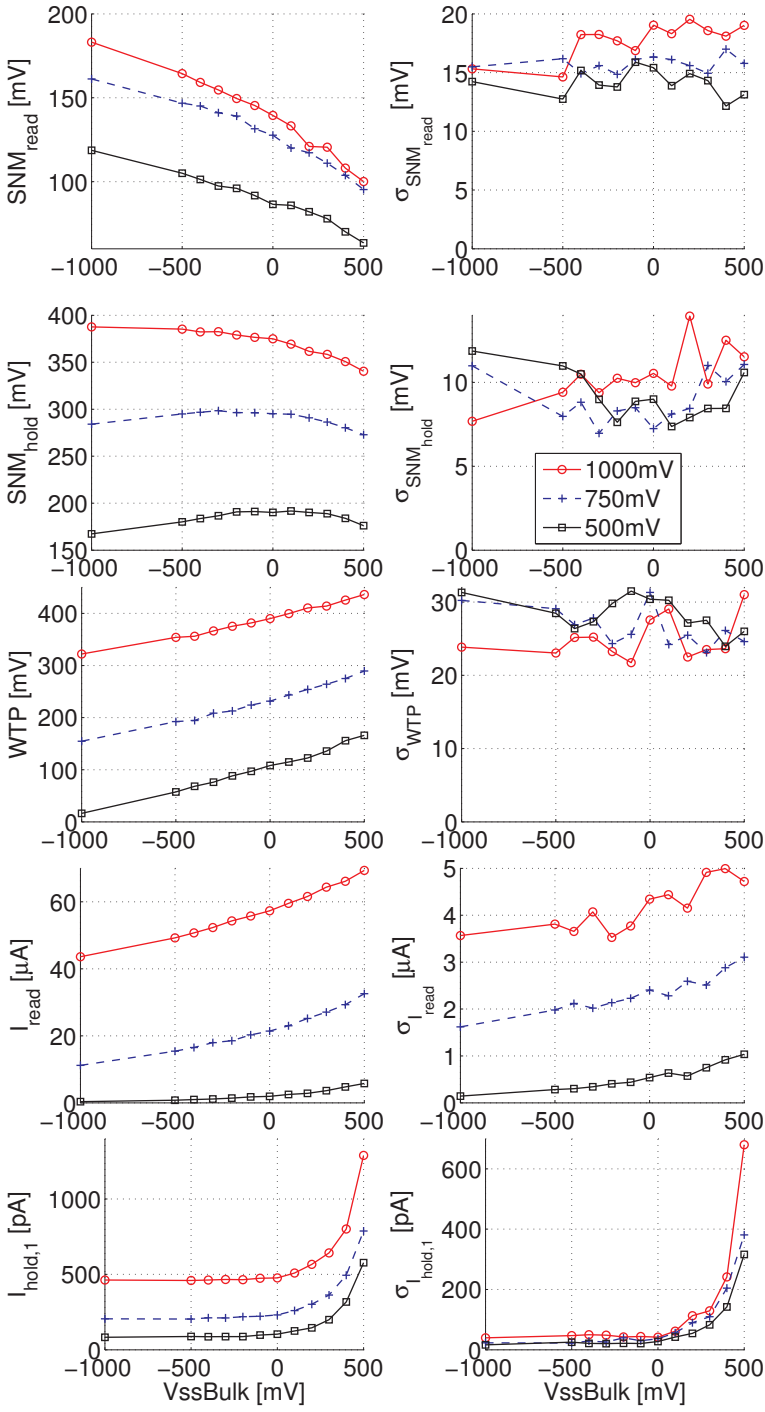


Figure 2.51: Mean and standard deviation for an exploration of the impact of $VssBulk$, while $VssCell$ is maintained at $0V$. Data are provided for $VddCell = VddWL = VddBulk = V_{BL,prech} = 1V$, $750mV$ and $500mV$. All other parameters are as in figure 2.36.

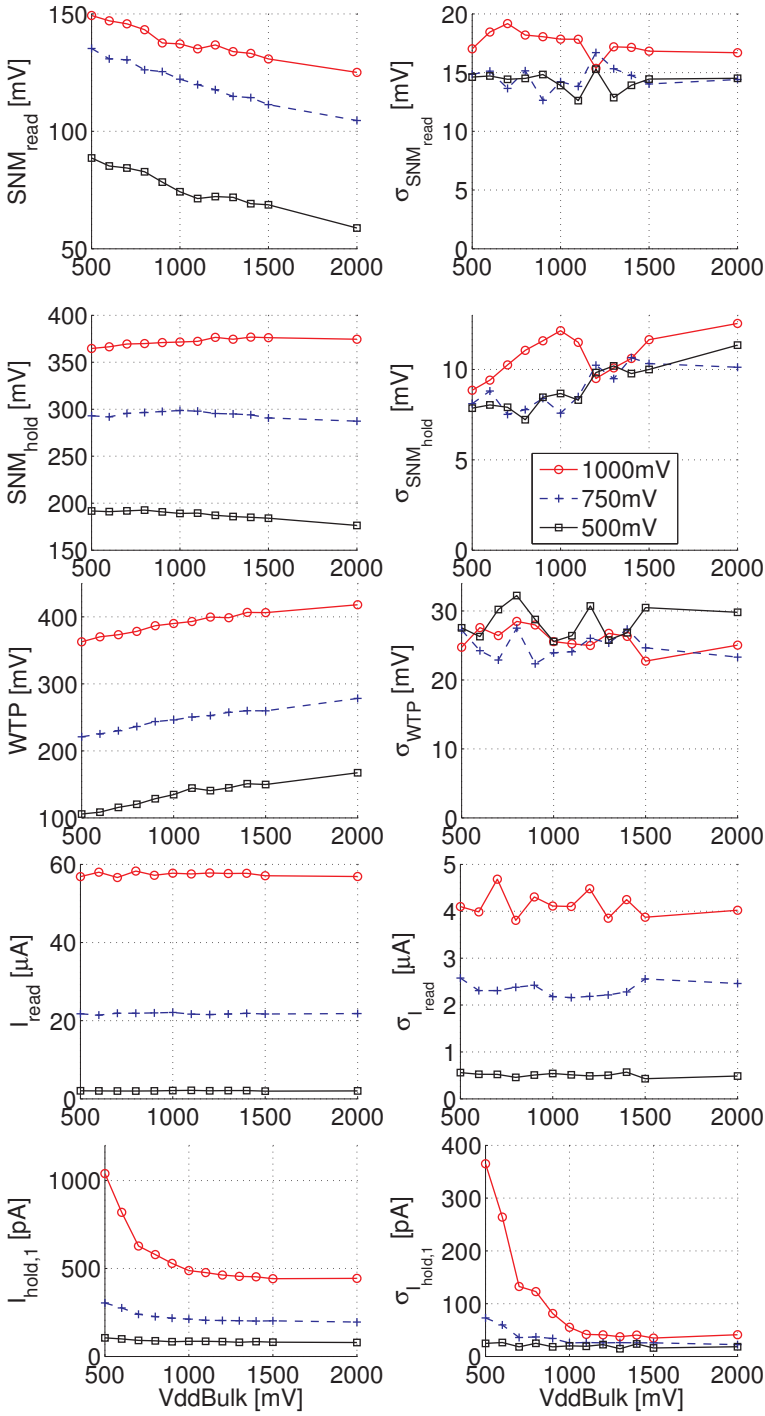


Figure 2.52: Mean and standard deviation for an exploration of the impact of $VddBulk$. Data are provided for $VddCell = VddWL = V_{BL,prech} = 1V$, $750mV$ and $500mV$. All other parameters are as in figure 2.36.

pull-down transistor in the cell read path, but it also increases V_{trip} of the cell invertors. For this cell, the net result is a relatively large increase in SNM_{read} at the cost of some I_{read} and WTP. When a positive $VssBulk$ is applied, it is termed forward body bias. In this case the junction diodes are forward biased so their leakage increases. However, up to $500mV$ (at room temperature), the leakage increase is probably acceptable if it occurs only during accesses.

Figure 2.52 discusses the impact of $VddBulk$, which affects the threshold voltage of the pull-up transistors. Values of $VddBulk$ above $VddCell$ result in reverse body bias, which increases $|V_{Tp}|$ and hence weakens the transistor. Values of $VddBulk$ below $VddCell$ result in forward body bias, which strengthens the pull-up transistors. In this technology, the impact is rather limited.

2.5.5 Impact of bit line precharge voltage

Figure 2.53 shows the impact of the bit line precharge voltage $V_{BL,prech}$. It is generally accepted that the ideal precharge voltage for read is around $VddWL - V_{Tn}$. Below this voltage, the cell side that stores a "1" gets discharged, which has a disastrous effect on SNM_{read}. This effect is very clear in the figure. Variations in the threshold voltage of the access transistor clearly move the ideal precharge voltage to a higher voltage range. During data retention, a lower $V_{BL,prech}$ reduces static power consumption somewhat.

2.5.6 Combined impact of Vdd_{cell} and Vdd_{WL}

Although figure 2.54 does not introduce significant new information, this plot is very useful to judge the combinations of $VddCell$ and $VddWL$ for which the cell is correctly read and written.

2.5.7 Practical design options

As discussed in section 2.5.1, the traditional 6T cell from figure 2.36, with high threshold transistors with minimal length and $200nm$ width is operational with a cell failure rate below or around 10^{-9} (in the Typical Typical process, so without considering inter-die variations) down to $VddCell = VddWL = 644mV$. Its read operation is slightly more problematic than its write operation, which can be resolved by e.g. increasing the cell β ratio a bit by increasing Wpd. The cell failure rate during data retention stays below 10^{-9} for supply voltages down to $VddCell = 235mV$. However, the soft error rate at such low supply voltages should be analysed before actually using such low voltages for data retention.

If lower supply voltages are desired or if a technology with larger variations is used, the cell design or operation mode has to be changed to obtain sufficient yield. The most straightforward solution is to increase the size of transistors to decrease the standard deviation of their parameters, and hence the standard deviation of the metrics. If the area of all transistors is multiplied by a factor M, all σ values reduce with a factor of $\sqrt{M}$. For example, with $M = 2$, all σ values are divided by $\sqrt{2}$. The standby voltage can then be reduced

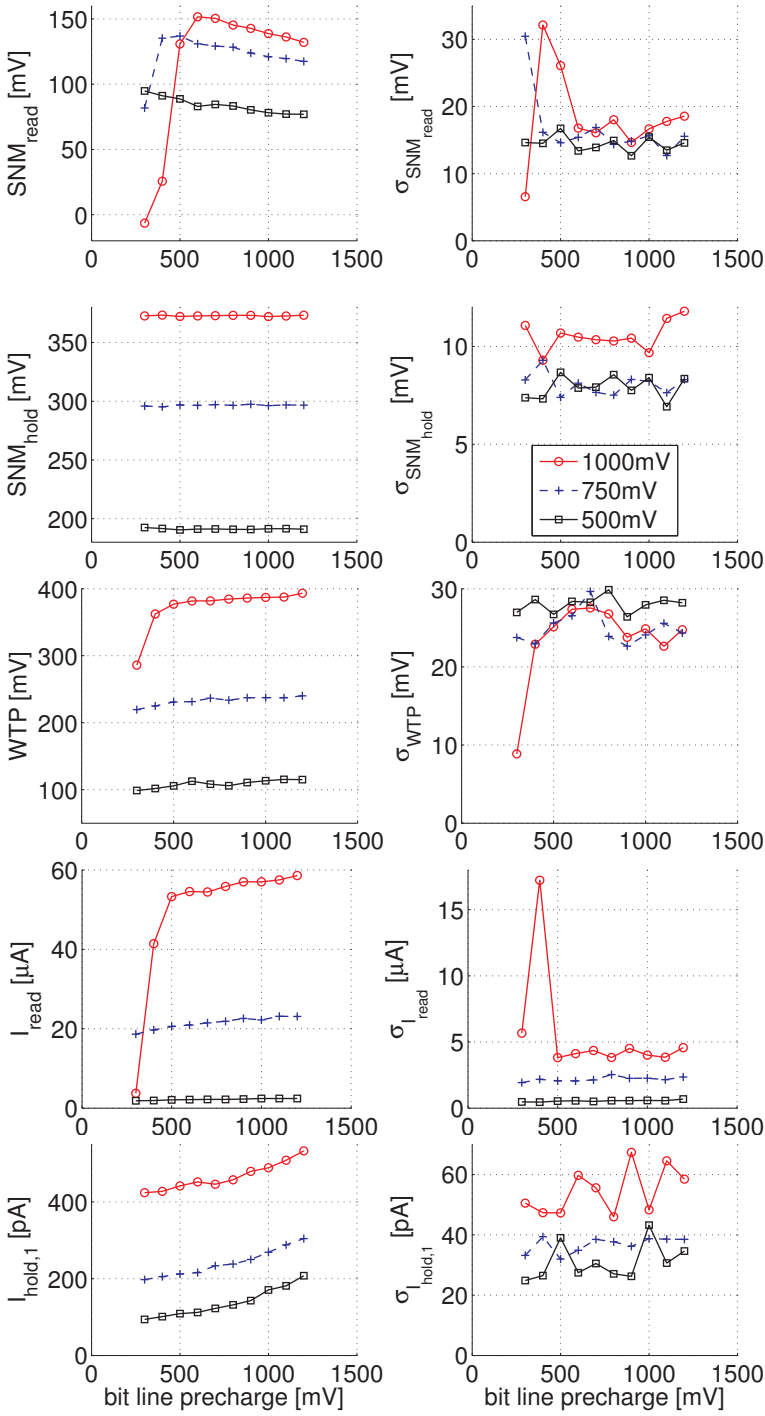


Figure 2.53: Mean and standard deviation for an exploration of the impact of the bit line precharge voltage $V_{BL,prech}$. Data are provided for $VddCell = VddWL = VddBulk = 1V$, $750mV$ and $500mV$. All other parameters are as in figure 2.36.

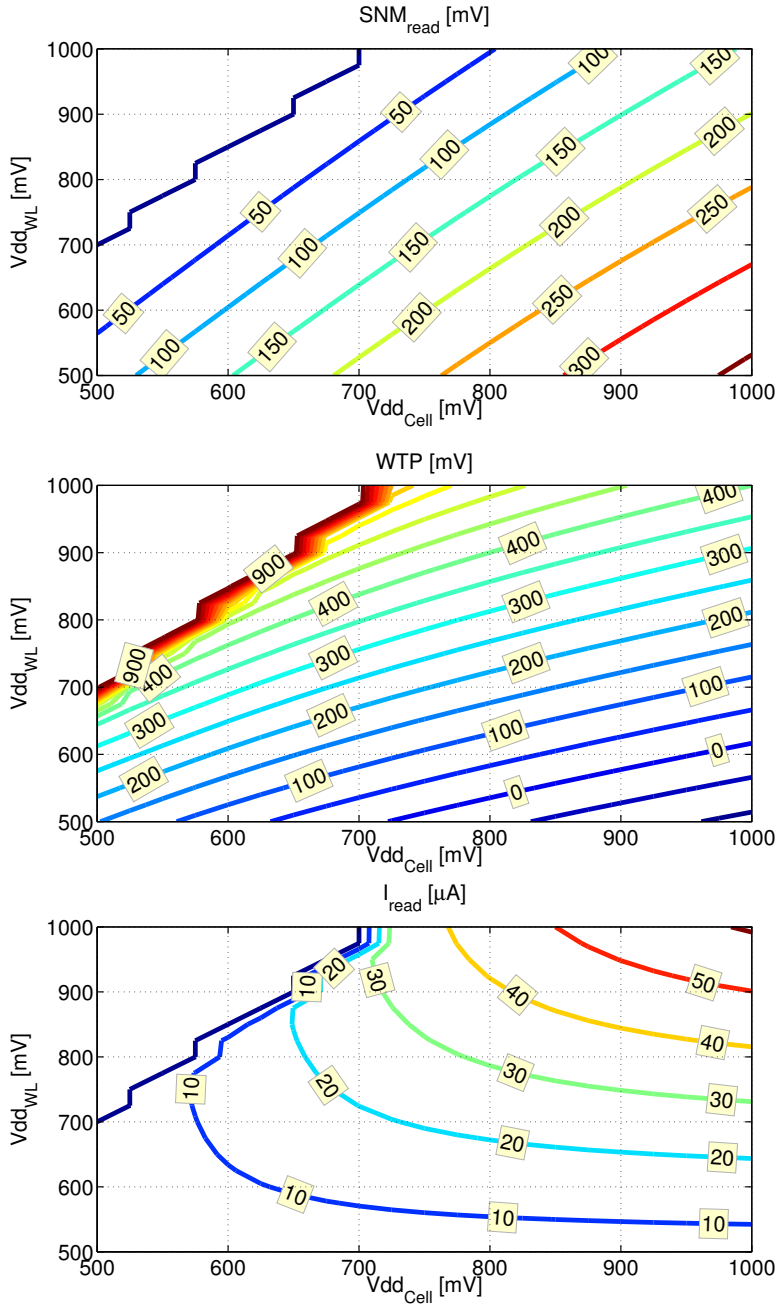


Figure 2.54: Performance metrics SNM_{read}, WTP and I_{read} as function of $VddCell$ and $VddWL$. ($V_{BL,prech} = VddBulk = 1V$; all other parameters are as in figure 2.36)

from $235mV$ to $186mV$, WTP is sufficient at $Vdd = 503mV$ instead of at $Vdd = 600mV$ and SNM_{read} is sufficient at $Vdd = 442mV$ instead of at $Vdd = 644mV$. Obviously, the nominal read current at the lower supply is significantly lower. However, increasing transistor sizes in the cell has a detrimental effect on overall memory performance. The size of the memory increases drastically, which increases the energy consumption and delay of the global interconnect and the cost of the memory. The cell leakage should also be considered carefully when cell transistors are increased. Larger transistors have proportionally more gate leakage. Wider transistors have proportionally more sub-threshold and junction leakage. The impact of longer transistors strongly depends on the specific technology, as the reduction of V_T with transistor length could prove dominant. Increasing the transistor size has a visible but limited impact on the available margins but has a large negative effect on overall memory performance, so it should be considered as the option of last resort.

Changes to the cell topology can significantly increase the cell operational margins. Some interesting examples will be discussed in section 2.7.

The available margins can also be increased significantly by using different voltages for read and write operations. For example, a read operation with $VddCell = 1V$ and $VddWL = 750mV$ rather than $VddCell = VddWL = 1V$, increases the mean SNM_{read} from about $130mV$ to $270mV$, as can be found in figure 2.49. During write, the combination of $VddCell = 750mV$ and $VddWL = 1V$ improves the WTP to almost $600mV$ compared to $WTP = 400mV$ at $VddCell = VddWL = 1V$. With such bias schemes, the 6T cell operation is assisted by the periphery. Such assistance allows the use of small 6T cells with rather poor intrinsic specifications and it enables the continued usage of traditional 6T cells in technology nodes in which traditional periphery cannot provide yielding minimal size memory cells. Some practical assist schemes that have already been described in the literature will be discussed in section 3.4.

2.6 Layout and wire routing

As the area of the cell must be minimised far beyond what the normal logic DRC rules allow, the layout of SRAM cells has always been a matter for technologists and lithography experts. However, a circuit designer needs to be aware of the implications of his circuit-level decisions in order to make reasonable decisions. With the increasing importance of transistor parameter variations in DDSM SRAM cells, optimisation of the cell requires more interaction between circuit designers, technologists and lithography experts, as was demonstrated in [Ver08a]. Therefore, a short introduction is provided here.

A brief but good overview of different cell layouts can be found in [Ish98]. Figure 2.55 shows a typical traditional SRAM layout. This layout can be completed with 2 metal layers without cell area increase. The word line runs in poly over the cells. To limit the RC delay of the word line, most often a parallel metal word line is provided as well and the two are connected every so many cells. There are

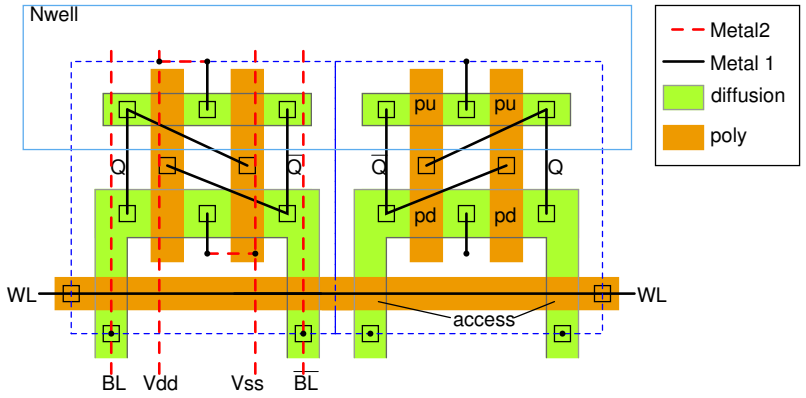


Figure 2.55: Artist's impression of the traditional SRAM layout style.

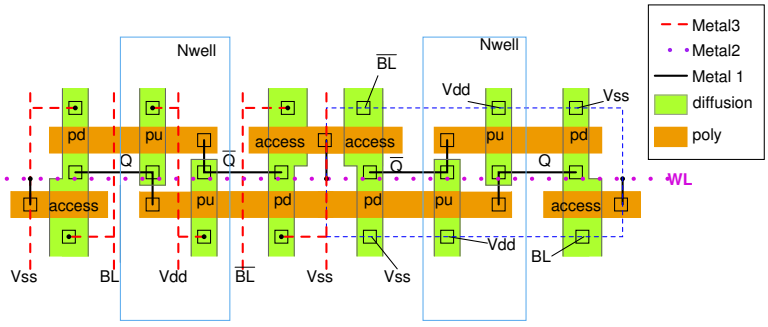


Figure 2.56: Artist's impression of the now generally used thin cell SRAM layout style.

a few problems with this cell. It is difficult to print lithographically due to the fact that poly lines run in both directions and due to the complicated connection between Q and $QBar$. The different poly orientation between the pull-down and access transistor can introduce significant systematic mismatch between these critical devices. In practice, even with optical proximity correction (OPC), the actually printed diffusion regions are not straight, but form a more or less smooth transition region. With these deformed shapes, overlay errors between the diffusion layer and the poly layer translate in different effective transistor widths, which again introduces systematic mismatch between the pull-down and access transistors. For this cell layout, horizontal overlay errors have an opposite effect on the two cell halves. This further worsens the problem. The cell ratio is not very good: the height of the cell, which defines the BL wire capacitance is typically larger than its width, which has less impact on memory performance. Finally, there is a signal integrity problem: the bit lines of two adjacent cells run immediately next to each other, without proper shielding.

Although many other cell layouts have been considered, it seems that the layout as shown in figure 2.56, sometimes called the "thinCell" layout, has become the de facto cell layout style. It was already mentioned in [Ish98] and it was used in a real design for the first time in [Osa01]. The version as shown here, with only vertical patches of diffusion was called a straight line "thinCell" layout in [Cha05a]. This cell has several advantages: it is easier to print and all transistor gates are parallel, which reduces systematic mismatch. Horizontal overlay errors no longer affect the effective transistor parameters. In this cell, with a different width for access and pull-down transistor, vertical overlay errors still cause systematic mismatch. If both transistors have equal width, this problem is resolved as well. All bit lines are now properly shielded and the cell ratio is very good. This results in a small wire capacitance for the bit line, both because it is short and because it allows a relaxed wire spacing. A disadvantage is that this cell requires 3 metal layers to enable a compact layout, unless special technology options are available, e.g. direct contacts from diffusion to poly.

With compact cell layouts, routing and contacting the signal wires and power supply wires is challenging. There are some requirements with respect to the orientation of different wires. It is clear that the word line and the bit lines must run orthogonal. As usual, we assume that the word lines run horizontally and the bit lines vertically.

Supply wires must be able to conduct all cell currents of all accessed cells. Therefore, power supplies are generally routed vertically, parallel with the bit lines. Even so, the voltage drop over this supply wire must be carefully considered. If a supply wire is routed along the word line direction, it has to sink the current of all accessed cells. An example will show that this will often be problematic. Assume a local supply wire that is $200nm$ wide which spans 32 accessed cells, with a cell width of $2\mu m$ and a cell current of $100\mu A$. The wire resistance is $0.1\Omega/\square$. The wire is $\frac{32 \cdot 2\mu m}{200nm} = 320\square$ long, so its resistance is $R = 320\square \cdot 0.1\Omega/\square = 32\Omega$. The total current is $I = 32 \cdot 100\mu A = 3.2mA$.

If all current would go through the entire wire resistor, the voltage drop over this local wire would be $\Delta V = R \cdot I = 102mV$. With the actual current distribution, the peak drop is about $50mV$. Apart from this voltage drop, the high current density can introduce reliability concerns due to electromigration. In the example $90nm$ technology, maximal allowed DC current is $2mA/\mu m$, so it should not be a problem yet. The voltage drop might or might not be an issue. However, longer supply wires should certainly be routed along the bit lines. Grid-connected supplies are possible as well. In the specific situation of sub-threshold memories, the cell currents are much smaller, and the supply wires are less problematic. In such memories, horizontal supply wires can be useful, as in [Ver08b].

Control wires which should connect only to the cells of the accessed word must run horizontal. Sometimes, control wires can be shared with non-accessed cells, for example the loop-cut signal in [Tak06] that avoids destructive reads. If they connect to two rows, they can be routed on the edge between two cells. If they connect to all cells of a local sub-matrix, the routing direction is free.

2.7 Alternative cell types

In section 2.5, it was shown that in a typical 90nm technology node, the traditional 6T SRAM cell with all transistor widths $200nm$ operates correctly under variations down to $Vdd_{Cell} = Vdd_{WL} = 650mV$.

In smaller technology nodes, variations will increase. It seems possible to design small, stable 6T SRAM cells in the traditional way in a 32nm node at $1V$ supply [Ver08a, Cha05a], but it will be on the edge. By employing circuit techniques to differentiate between read and write, e.g. a higher word line voltage during the write operation, the operational margins can be extended significantly, which should enable continued use of the 6T cell. Some practical implementations of such differentiated read and write operations will be discussed in 3.4.

However, it is not because it is possible to further scale the traditional 6T SRAM cell, that this cell type is also the best choice for all future designs. This section discusses some alternative cell types that have been presented in the literature.

2.7.1 8T-SRAM cell (readbuffer)

Figure 2.57 shows the schematic of the 8T-cell as presented in [Cha05a]. Write and data retention is performed by a traditional 6T SRAM cell with $\beta = 1$. The read operation is performed by a separate readbuffer. This cell requires a read word line and a write word line. In [Cha05a], they seem to fit in the existing cell height of the "thinCell" layout. The cell uses complementary write bit lines and a single-ended read bit line. In theory, it would be possible to share the read bit line with the writeBar bit line, but that would not be wise. There is space enough to route them both, and sharing would increase the capacitive load of the read bit line. In [Cha05a], the area overhead as compared to a 6T-cell with $\beta = 2$, is estimated at 30%.

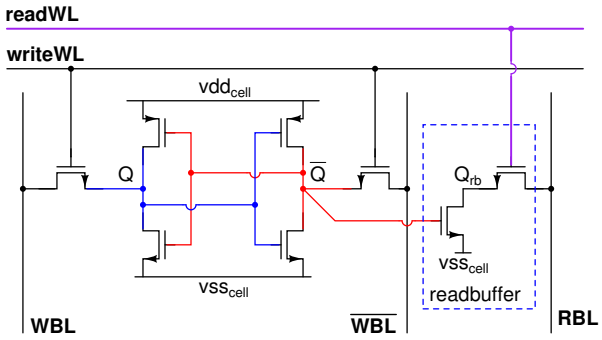


Figure 2.57: The 8T SRAM cell: a 6T cell for write and data retention and a 2T readbuffer.

This cell eliminates the read stability problem entirely. However, to be of any benefit, the write word line should not be shared between multiple words. This would lead to half-selected cells, cells that should retain their data, but with the write word line asserted and both write bit lines precharged high. This is exactly the situation that causes read stability issues in normal SRAM cells.

This cell can be optimised separately for read and write. The 6T part can be designed specifically for write and data retention: a β of 1 at most and a very high threshold voltage and thick oxide for the transistors in the cross-coupled invertor pair. This improves the write-ability and reduces the static leakage. The 2T readbuffer can be optimised for read speed, without any risk for read stability and write-ability issues. With proper power gating of this readbuffer, it should be possible to use transistors with much lower threshold voltages in this buffer without increasing the static power consumption during standby significantly. Gate leakage might be an issue though.

If the next stage is a sense amplifier, one should keep in mind that the cell read current for the $8T$ cell is either $+I_{buffer}$ or 0, while that of a traditional 6T cell is either $-I_{cell}$ or $+I_{cell}$. In this case, the relevant read current comparison is between I_{cell} and $\frac{I_{buffer}}{2}$.

If full-swing local read bit lines are used, a single-ended read operation can reduce the energy consumption as the LBL activity is about halved. For the impact of a single-ended read operation on the energy consumption of the global bit lines, see chapter 6.

This cell is certainly a simple and elegant solution for cases where the traditional 6T cell does not meet the specs. The main question is whether it is the best available solution for the given area increase (30% compared to a $\beta = 2$ cell, about 40% compared to a $\beta = 1$ cell).

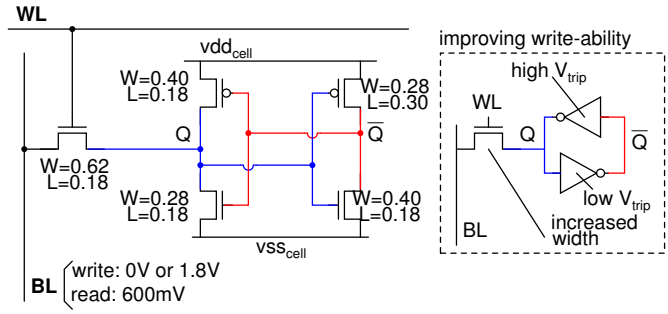


Figure 2.58: The 5T-SRAM cell from [Car04].

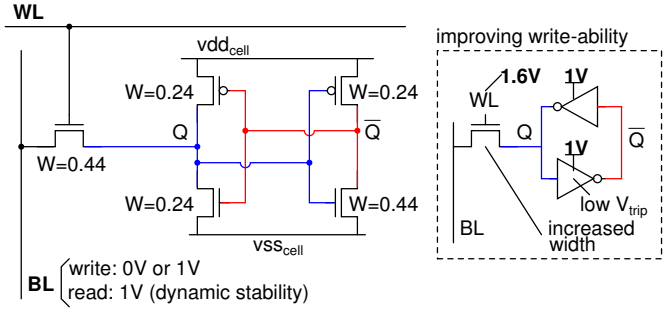


Figure 2.59: The 5T-SRAM cell from [Cos07].

2.7.2 Single-ended cell

In earlier technologies, the 6T cell provided plenty of margins. In those days, several attempts have been made to reduce the number of transistors in the cell while still providing adequate margin. Here, only the single-ended cells are discussed. Figure 2.58 shows the cell that was used in [Car04] in a $0.18\mu m$ technology. In theory, such a single-ended cell has two benefits: it has one transitor and one bit line less than a 6T cell, so its area should be smaller. The main benefit is that only cells that store a "0" introduce activity on the bit lines during read. This can be a significant advantage.

In a 5T cell, the read and write operation must complete through a single access transistor. This can only work if there is a difference in the bias signals that are applied between read and write "1". Otherwise, both operations would be identical. In [Car04], the bit line precharge voltage for read was set to $0.6V$, while for write "1", it was forced to $1.8V$. A stronger difference could have been realised by changing $VddWL$ between both operations. In our first design, we used the 5T cell from figure 2.59 [Cos07]. Here, read stability was realised using the novel concept of dynamic stability. This concept will be discussed in section 3.6. Essentially, the small capacitance of the local bit line is discharged during read before the cell can be upset.

A more fundamental problem of single-ended cells with a NMOS access transistor is the difficulty to write a high value to the cell. As the voltage on node Q rises, the access transistor becomes weaker due to the reduced $V_{gs,access}$, while $V_{gs,pull-down}$ remains at Vdd. This is especially problematic for cells with high threshold transistors and at reduced supply voltages. In the cell from figure 2.58, this problem was addressed by significantly increasing the width of the access transistor as compared to that of the pull-down transistor and by selecting V_{trip} to aid this operation. In [Cos07], we used a similar sizing scheme. More importantly, we had to use $VddWL = 1.6V$ and $VddCell = 1V$ to aid the write operation to obtain acceptable margins.

Other solutions are possible. In [Tra96], $VssCell$ at the side of node Q was made floating during write to significantly weaken the pull-down transistor. $VssCell$ at the side of node $QBar$ must be kept low as otherwise V_{trip} would increase as well.

In our second design, we resorted to an even more drastic solution. The cell in figure 2.60 has an additional transistor in its main pull-down path. During write, this transistor is disabled, which allows the access transistor to increase the voltage at node Q to about $VddWL - V_{Tn}$, which was sufficiently above V_{trip}.

In [Zha08], the cell of figure 2.61 was used in a sub-threshold memory. It uses complementary access transistor, and to improve write-ability, the cell feedback invertor is gated during write. A main disadvantage of such complementary access transistors is that it increases the bit line capacitance.

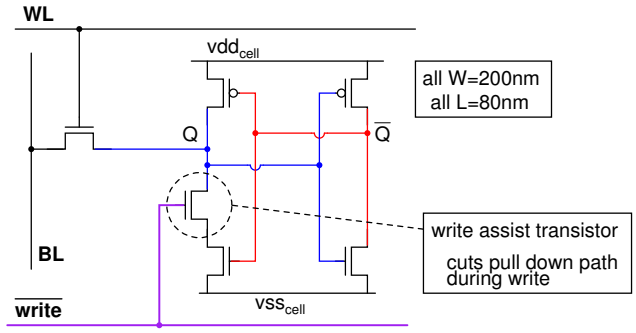


Figure 2.60: The single-ended SRAM cell from [Cos08].

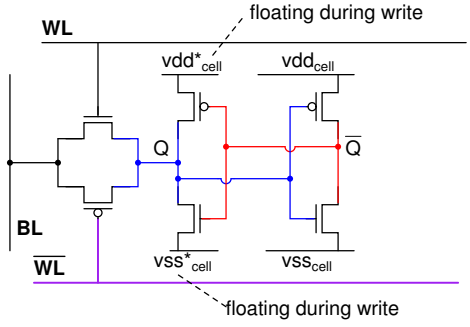


Figure 2.61: The single-ended cell from [Zha08].

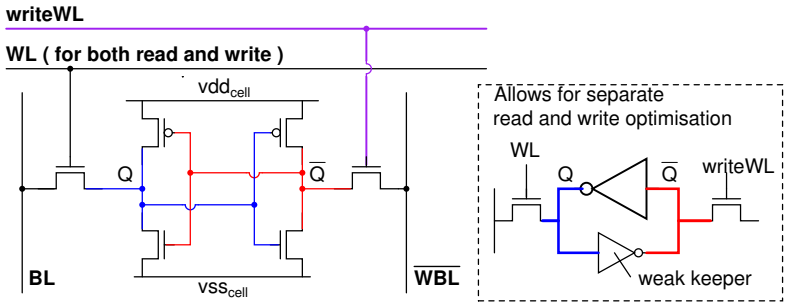


Figure 2.62: A cell with single-ended read and double-ended write.

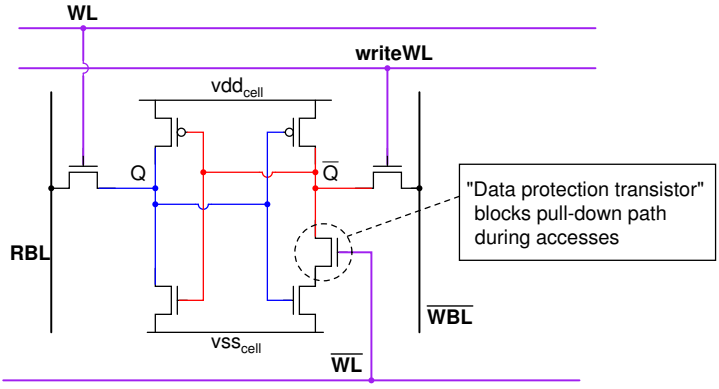


Figure 2.63: The read-static-noise-margin-free 7T cell [Tak06].

2.7.3 Single-ended read, double-ended write cell

A single-ended read looks attractive, but a single-ended write is very problematic. Figure 2.62 shows a cell which uses only one access transistor during read, but both access transistors during write. Such a cell was introduced in [Pig01]. The cell in [Tak06] (figure 2.63) uses a similar scheme, but with a significant addition. This provides the advantages of a single-ended read in terms of activity, without the problems of a single-ended write. But there is more: as one side of the cell does not have to support the read operation, it can be optimised specifically for write and data retention. The right cell halve can have a very weak invertor, for example with a higher threshold voltage and hence less leakage, combined with a stronger access transistor. This way, this cell is better write-able than a traditional 6T cell. If the routing for the 8T readbuffer cell indeed fits in the cell height of a “thinCell” layout, this should also be true for the additional write word line.

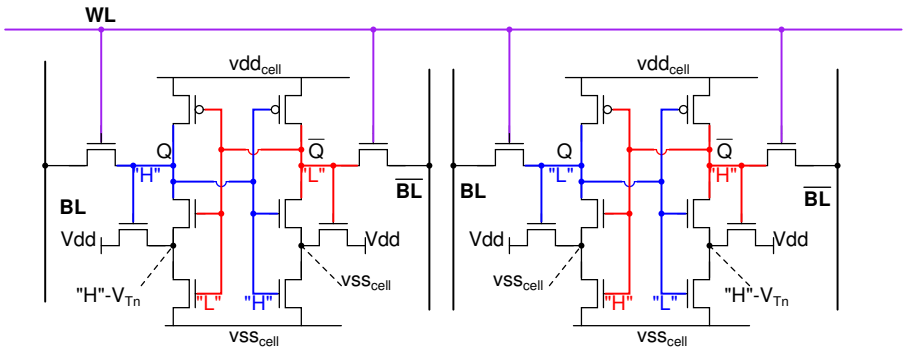


Figure 2.64: A Schmitt-trigger based cell [Kul07].

2.7.4 Read-static-noise-margin-free 7T-cell

The read-static-noise-margin-free cell from figure 2.63 was introduced in [Tak06]. Notice the singled-ended read and double-ended write. During data retention, $\overline{WL}$ is high, and there is no essential change in behaviour compared to the 6T cell. Because the cell is less balanced, SNM_{hold} is reduced though. During read, $\overline{WL}$ goes low, and only the access transistor at the side of the read bit line is enabled. With a high precharge on the read bit line, $SNM_{read,1}$ is not a relevant problem. The critical situation for stability occurs when Q stores "0". When the voltage on node Q rises above V_{trip} of the cell invertor, node QBar discharges, and data is lost. In the cell from figure 2.63, the data protection transistor makes it impossible for QBar to discharge. This essentially increases V_{trip} to a very high value. It is however impossible to correctly analyse this cell with static analysis methods such as SNM_{read}. It is not clear how the authors could provide butterfly curves for the read operation.

This cell indeed resolves the SNM_{read} issue, and does this without impacting I_{read} as there are no changes to the pull-down path. However, this requires an additional transistor and the routing of three word line control signals (WL, $\overline{WL}$ and $writeWL$). Even though $\overline{WL}$ can be shared with non-accessed cells, it is not really clear whether the overhead will be significantly smaller than that of the 8T readbuffer cell.

2.7.5 Hysteresis cell

SNM_{hold} of the traditional 6T SRAM cell is limited to $Vdd/2$. Figure 2.64 shows a cell which employs feedback to enable a hysteresis effect in the transfer curves of the cell invertors. It was intended for sub-threshold operation. If Q stores a high value, the left pull-down transistor stack is weaker due to the increased voltage at the internal node. This effectively increases the cell SNM_{hold} beyond $Vdd/2$. According to [Kul07], it also reduces the standard deviation of SNM_{hold}, but the plotted distributions are certainly not Gaussian. The write trip point of this cell is improved as well, as V_{trip} of the invertors is higher than

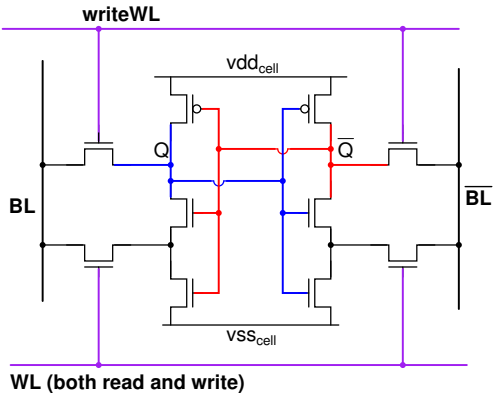


Figure 2.65: Another Schmitt-trigger based cell [Kul08].

in a normal 6T cell. The read stability is a slightly more complicated story. Assume Q="0". Due to the hysteresis effect of the invertors, V_{trip} is higher, and a higher voltage increase on node "Q" can be tolerated. However, due to the stack of two transistors in the pull-down path, the voltage increase for the same I_{read} will be higher as well.

The cell in figure 2.65 solves this last issue [Kul08]. However, this cell does not improve SNM_{hold}, adds an additional control wire and increases the bit line load.

2.8 Summary

This chapter thoroughly discusses the memory cell. Considerations regarding the peripheral circuits that connect to it are avoided as much as possible. These peripheral circuits and their interaction with the memory cell are described in depth in the next chapter.

Section 2.2 outlines the most important properties of a memory cell. The contribution of the cell to the bit line capacitance, $C_{BL,cell}$ is a crucial cell property, certainly in memory organisations without bit line subdivision. It has a large impact on the read and write energy of the memory, and together with the cell read current I_{read}, it defines the cell read delay. Other important considerations are the static power consumption, the cell yield and reliability and the cell area. Even if cell area is not important from a cost point of view, it does increase wire lengths and hence wire capacitances, which increases the energy consumption and the delay of the memory.

Section 2.3 reviews the best known metrics to describe cell performance and stability. The behaviour during data retention is captured by the static noise margin SNM_{hold}, the data retention voltage DRV and the static power consumption P_{static}. Read stability is assessed with SNM_{read}, while the cell read delay is proportional to $\frac{C_{BL}}{I_{read}}$. The write trip point WTP, the highest bit line

voltage that is low enough to overwrite the cell data is a good metric for cell write-ability.

Section 2.4 outlines the available methods to obtain a good estimation of performance and yield under transistor variations. For most of these methods, the quality of the cell metrics that is used is of vital importance. The more smoothly the metrics change with changes in transistor parameters, the easier it is to estimate the distribution of these metrics.

Section 2.5 treats an example 6T cell design. For a failure rate $fr = 10^{-9}$ in the typical process point, a $90nm$ 6T cell with all SPHVT transistors with $W = 200nm$ and $L = 80nm$ requires a DRV of $240mV$ and an active supply voltage of $Vdd_{WL} = Vdd_{cell} \geq 650mV$. An exploration of the impact of changes in transistor dimensions and bias voltages on the cell metrics is provided, and practical design options are discussed. When different bias voltages are applied during read and write operations, the available margins increase significantly. With such bias schemes, the 6T cell operation is assisted by the periphery. Such assistance allows the use of small 6T cells with rather poor intrinsic specifications and it enables the continued usage of traditional 6T cells in technology nodes in which traditional periphery cannot provide yielding minimal size memory cells. Some practical assist schemes that have already been described in the literature will be discussed in section 3.4.

Section 2.6 briefly treats the layout of SRAM cells in DDSM technologies. Section 2.7 reviews some of the more interesting alternative cell types that have been proposed in literature. The 8T cell with readbuffer is interesting as it completely circumvents the read stability issue. Cells with single-ended read and double-ended write are interesting as well: one cell side can be optimised for read and the other for write, with a very small area overhead compared to a traditional 6T cell.

Contributions and highlights

This chapter discusses the analysis and design of SRAM cells in technologies with large intra-die variations. All required methods are available in the open literature, but this is the first text that brings everything together in a more or less accessible way.

SRAM cells have always been subject to tight and often contradicting constraints. They require a good read stability and must be easy to overwrite. They must provide a large read current, but must have a small area and a small bit line capacitance. With the increasing importance of leakage currents and transistor variations, the cell now seems over-constrained. One of the central themes of this work is the search for efficient ways to relax the cell requirements by improving the peripheral circuits.

Chapter 3

Local memory matrix

3.1 Introduction

In the previous chapter, the memory cell was studied as a stand-alone circuit. This chapter outlines how these cells are connected together to build up a local memory matrix and treats the immediate periphery that is needed to access such a local matrix. Techniques that ease the cell requirements receive special attention.

Section 3.2 describes the basic organisation of a local block.

Section 3.3 discusses the peripheral control circuits. In the traditional memory architecture, this is the local word line driver, but in many state-of-the-art-designs that are targeted at technologies with large intra-die variations or at ultra-low voltage operation, additional control wires are introduced, as described in section 2.7. Memories with subdivided bit lines use similar drivers to control the local bit line periphery.

Section 3.4 introduces an interesting and challenging set of new design techniques that is quickly becoming mainstream as technology moves beyond 65nm. To aid the cell operation, different bias voltages are applied for the different operation modes. The underlying effects have already been analysed in section 2.5. Section 3.4 also provides an overview of bias schemes that have been discussed in the literature and addresses some of the basic considerations regarding the different bias schemes.

Section 3.5 describes different implementations for the bit line hierarchy, with the local bit line as focal point. This section shows that a good bit line hierarchy can significantly reduce the impact of the cell current I_{read} and the cell's bit line capacitance $C_{BL,cell}$ on the access time and energy consumption of the memory.

Section 3.6 introduces the concept of dynamic read stability, one of the major contributions of this work. With the introduction of short buffered local bit lines, the traditional metric for cell stability during the read operation, SNM_{read}, becomes an excessively pessimistic estimate. This section first shows that cells with large negative SNM_{read} can still be read without causing a read upset. A novel metric, SNM_{tran}, is introduced to quantify a cell's dynamic read stability as a value which can be compared directly with SNM_{read} and SNM_{hold}. The section concludes with an overview of properties of the cell periphery that influence the cell's dynamic read stability: the number of cells per bit line, the word line rise time and the bit line precharge voltage.

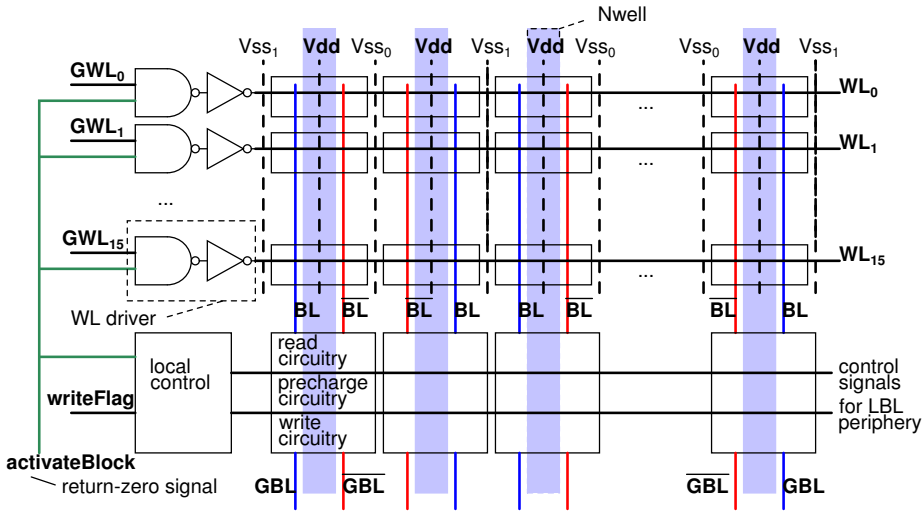


Figure 3.1: A typical floorplan of a local memory block. In normal designs $Vss_1 = Vss_0 = Vss$.

3.2 Local block organisation

Figure 3.1 shows a typical floorplan of a local memory block that uses the traditional 6T thin cell layout from figure 3.2. This layout style was discussed in section 2.6.

When a word that is stored in this local block is accessed for read, one of the word lines is pulled high, which enables the access transistors of all the cells on the corresponding row. The bit lines of the local block have been precharged to a relatively high voltage before word line activation. At lower bit line voltages, read upsets will occur because the high cell side gets discharged through the

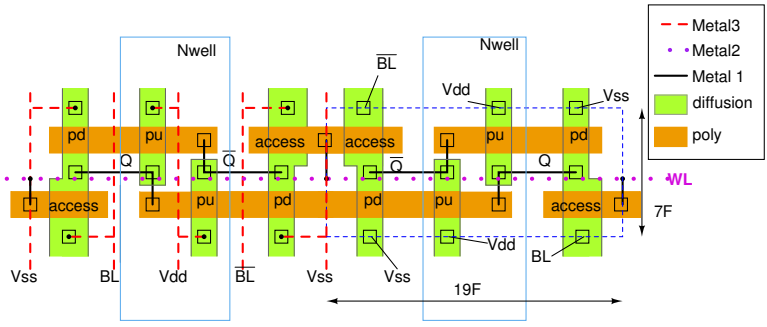


Figure 3.2: Artist's impression layout of a 6T cell with "thinCell" layout. Typical cell dimensions for litho-optimised cells ($19F$ by $7F$) are indicated. Cells that obey logic DRC rules are about $31F$ by $10F$.

access transistor (see section 2.5.5). The cell side that stored a 0 then develops a voltage drop $\Delta V(t)$ on the associated bit line,

$$\Delta V(t) \;=\; \int_{t_0}^{t} \frac{I_{read(t)}}{C_{BL}(t)} dt \;\approx\; \frac{I_{read} \cdot \Delta t}{C_{BL}}$$

If the voltage transition on the word line is steep, the approximation is acceptable as both C_{BL} and I_{read} are rather insensitive to the bit line voltage in the region of interest, e.g. above $Vdd/2$. The bit line periphery must transfer and potentially amplify the signal on the local bit lines towards the global bit lines.

When a word of the local block is accessed for write, the bit line periphery pulls one of the bit lines of each bit line pair low and keeps the other one high. The word line is activated, and all cells on that row for which one bit line is pulled down, are overwritten.

If the word line is fully subdivided, all cells that are connected to the activated word line belong to the word that must be read or written. This is the most simple matrix organisation and because no cells are needlessly activated, this organisation typically results in the lowest energy consumption as well. However, as each word requires its own word line driver and each local bit line pair requires its own periphery, the area overhead can be large, especially when the word length is very short. The global bit line pair must also be routed in a single cell pitch. A second potential problem is that the cells of a word are physically adjacent, which increase the chance of multiple bit errors within a single word due to radiation events (soft errors, as discussed shortly in section 2.2.10). This is only relevant in systems which incorporate error correction or detection codes, as the codes can typically only repair one error in each word.

Figure 3.3 shows a local matrix organisation with a partially subdivided word line: each word line connects to two words. During an access, there will be activity on all bit line pairs. However, only the activity on one of both bit line pairs is actually desired. Only this information should be transfered from or to the global bit lines, so it should be possible to share part of the peripheral circuits and the global bit lines. In practice, it turns out to be rather complicated to find a good way to implement such shared periphery, so only minor gains should be expected.

Organisations in which one word line connects to multiple words, introduce many problems. The main problem is the so-called half select problem during write. The cells on the activated rows that should retain their data, have an activated word line, but both bit lines stay precharged rather than that one is pulled low. In traditional memories, this is exactly the same situation as during read, so no stability problem occurs. However, in memories in which the word line voltage is lowered during read operations to improve read stability or increased during write operations, the half-select situation would result in data loss. A read-modify-write scheme in which the data of the cells that should be retained is first read and then written back by the local periphery could solve this issue, but at a rather high control overhead. Without such a scheme, the

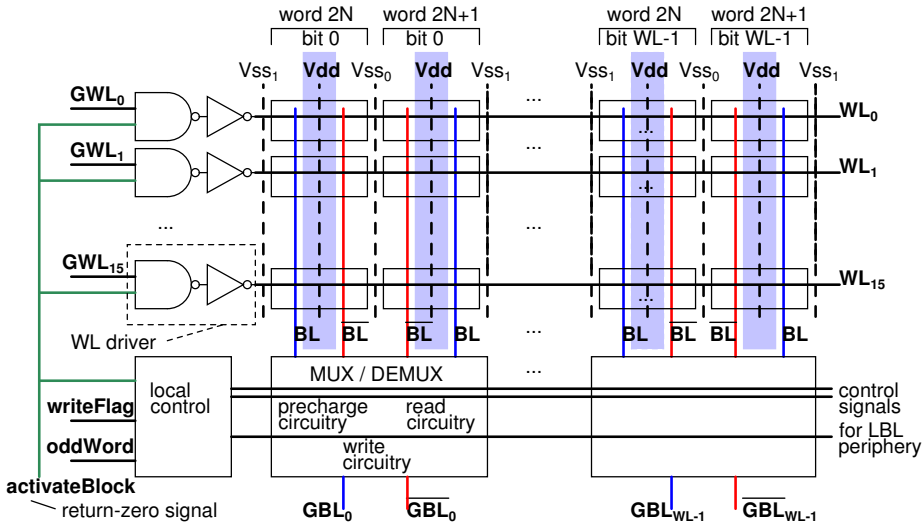


Figure 3.3: A typical floorplan of a local memory block, with partially subdivided word line: each word line connects to two words. WL is the word length.

word line voltage can not be used as a differentiator between read and write unless the word line is fully subdivided.

In some cell layout styles, cells of different words can be arranged in an interleaved fashion on a single row by routing two or even more word lines over each row of cells, as shown in figure 3.4. The setup from figure 3.4a was used in our $0.18\mu m$ design. By sharing a bit line between two adjacent cells, the effective bit line wire length per cell is halved. However, all cell capacitance is still connected to this single bit line. The setup from figure 3.4b was used in our $90nm$ design. This structure matches the structure that was represented in figure 3.3. If it would be possible to share the bit line periphery effectively between bit lines A and B, this would reduce both the wire capacitance and the number of cells that are connected to a single bit line for a given area overhead of the bit line periphery. However, as mentioned before, sharing this periphery in an effective way is difficult. In the thin cell layout style, two adjacent cells share a word line contact, so it is impossible to interleave cells in this way.

In technology nodes down to about $90nm$, the choice of the cell layout is more or less free. Beyond $90nm$, the cell layout needs to be lithography friendly, and one should probably not deviate from the thin cell layout style unless there are very compelling reasons. Changes that are compatible with this layout style are more likely to be adopted. Using the thin cell layout style as shown in figure 3.2 has some implications on which types of control signals can be introduced at a reasonable cost. In the next section, no special technology options are assumed, so the metal layer choices of figure 3.2 are used.

First of all, one could wonder which additional wires would fit in the available

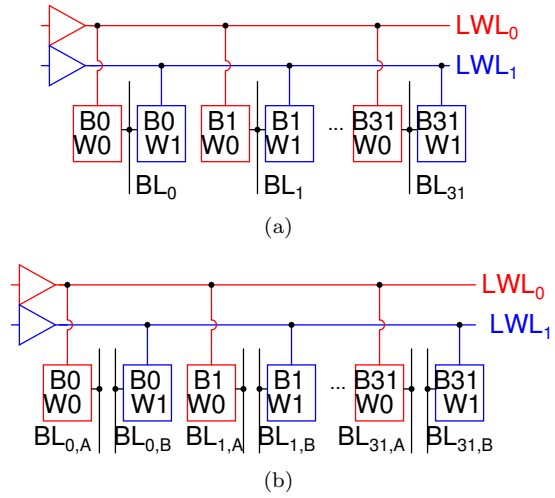


Figure 3.4: Two ways to interleave cells in a memory with fully subdivided word lines.

cell height and cell width ($7F$x$19F$ or $10F$x$31F$). Control wires in the direction of the word lines should be routed in M2, in parallel with the word line. The cell height is only $7F$. In this height, the word line and the vias for the supplies and bit lines along the top and bottom edges of the cell must already be accommodated. As the top and bottom edge should be counted only half each, this is the equivalent of two tracks. To add a second control wire requires the addition of a third track. With the typical wire pitch of $3F$ for M2, this would increase the area to $\frac{3 \cdot 3}{7} = 129\%$. However, with dedicated rules, it might fit. In cells which adhere to DRC rules, the cell height is about $10F$, and there should be no problem to fit an additional track in this direction. If a control signal could be shared between the cells of two rows, it should only introduce 0.5 additional tracks per cell height. However, due to the contacts on the cell's top and bottom edges, it might be challenging to route such a wire.

Control wires in the direction of the bit lines and power supply lines should be routed in M3. Typical cell width is $19F$, and BL, Vdd, $\overline{BL}$ and half of Vss_0 and Vss_1 should be routed in this width, 4 tracks in total, while up to 6 tracks should fit in this width. However, if this is done, the bit line wire capacitance will increase somewhat.

Bulk voltages and power supplies are shared for entire (local) columns. The finest control granularity is hence a local block. In interleaved layouts, it is possible to change $VddBulk$ and Vdd for the even and odd cells separately. It is always possible to change Vss_0 and Vss_1 separately. Vss_1 is the ground voltage for the Q side of the cell, Vss_0 is the ground supply for the $QBar$ side of the cell.

Two horizontally adjacent cells share a word line contact. However, the two

word line contacts of a single cell are separate, and the contact sharing occurs between like cell sides, both Q or both $QBar$. This keeps the option open to use a single-ended read, double-ended write.

The boundaries of the local matrix deserve some special attention. With many other cell layouts, the smallest area is obtained when the local matrix is mirrored around both the horizontal and the vertical axis to form a conglomerate unit of 4 local matrices, as this allows to share cell contacts at all joining edges. When this block is replicated, contacts and supply wires in the periphery can be shared as well. With the thin cell layout, the advantages in the periphery are still there. However, no cell contacts can be shared as each edge contains either a bit line or a word line contact, which must be unique for each local matrix. Some area will hence be lost at the edges of the local matrix, and the cell environment for cells at the edges and at the corners will be different from that in the center of the matrix. Some designs seem to distribute the bit line periphery between the top and bottom edge of the local matrix, which might increase the number of contacts that can be shared.

Because the cell environment is different for these cells at the edges, a systematic shift in their transistor parameters can occur. In analog design, this problem is often addressed by introducing dummy cells around the edge, but with small matrices of e.g. 16 by 32 cells, the area overhead would be huge. As the memory operation is not that sensitive to the transistor parameters, dummy cells can probably be avoided in all reasonable technologies. However, these systematic shifts will result in some additional margin loss. This margin loss can to some extend be limited, and part of the remaining loss can be predicted. One cause of systematic mismatch for edge cells is a difference in the effective patterns that is printed by photo-lithography. With proper optical proximity correction (OPC) applied to the combination of the edge and inner cells, this difference can be minimised, a technique that is already applied in some designs, e.g. [Kaw08a].

There is one more thing to say about figure 3.1. Ideally, the local block control circuitry should fit in the cross section of the word line drivers and local bit line periphery, which is really small. After mirroring to create the larger matrix, 4 of such control circuits are adjacent, so contacts and supplies can be shared. It seems much harder to share logic between the different blocks. In both our designs, quite a lot of control logic was used in the local block. In the $90nm$ design, especially the timing and the bit line control circuits required a lot of area. This forced us to extend their layout along the word line drivers and along the local bit line periphery. The bit line control was complicated due to the implementation of the sub local bit lines, while timing became involved because we wanted to derive all events from the rising and falling edge of a single return-zero signal, *activateBlock*. Based on this experience, it seems wise to seriously consider to limit the local control logic to the bare minimum, even if this requires additional external control signals for timing. The trade-off between local and global timing circuits is explored in section 7.3.2.

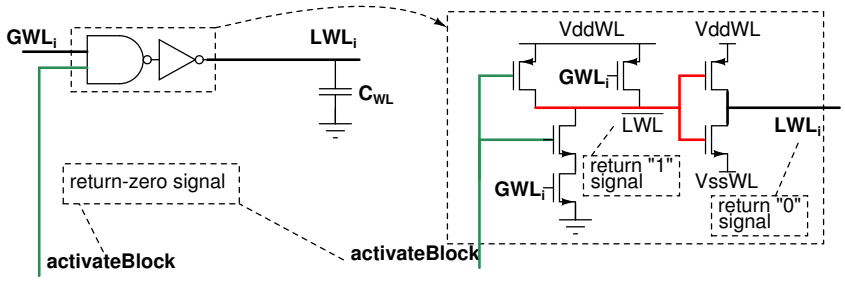


Figure 3.5: A basic word line driver.

3.3 Word line drivers

As shown in figure 3.1, a typical local word line driver performs two functions: it acts as the last combinational stage in the decoder (NAND gate) and it buffers the signal towards the word line (invertor). Other control circuits behave very similar to the word line drivers, so they are not discussed separately.

At least one of the inputs to the word line driver should be a return-zero signal, as it is impossible to correctly precharge the local bit lines when the word line is not disabled in between two accesses. In figure 3.1, the return-zero signal is *activateBlock*, a logical choice as this signal can be used to control the local bit line periphery as well.

Figure 3.5 shows a very basic CMOS word line driver that combines two input signals (GWL and $activateBlock$) to generate the word line signal LWL. The load on the word line consists of the gates of the access transistors and of the word line wire capacitance. For example, in a $90nm$ technology, with a traditional 6T cell and access transistors with $W = 200nm$ and with 32 cells per word line, $C_{wire} \approx 32 \cdot 2.5\mu m \cdot 0.2fF/\mu m = 16fF$ and $C_{gates} \approx 32 \cdot 1 \cdot 0.35fF + 32 \cdot 1 \cdot 0.2fF = 17.6fF$, as half of the access transistors will remain in sub-threshold operation ($C_g \approx 0.2fF$) and half of the access transistors will enter into the inversion region ($C_g \approx 0.35fF$). A more accurate estimation based on spice simulations gives a slightly lower value, $C_{gates} = 14fF$ if cell nodes and bit line voltages are forced with power supplies to their initial values, $12.5fF$ if the voltage increase on the internal cell node during read is simulated. Essentially, $C_{WL} = C_{gates} + C_{wire} \approx 14fF + 16fF = 30fF$.

This section now first discusses a basic word line driver, which is used as an example to highlight the important specifications of a word line driver. The remainder of this section discusses some interesting implementations for word line drivers.

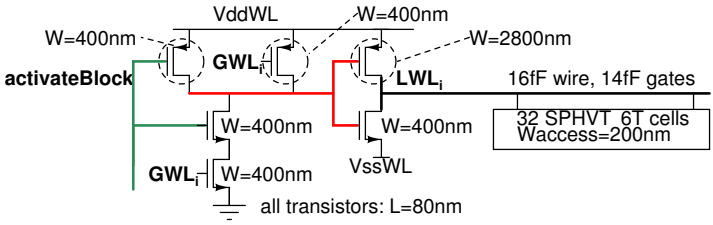


Figure 3.6: An example word line driver for a word line with 32 cells.

3.3.1 Example word line driver

3.3.1.1 Area

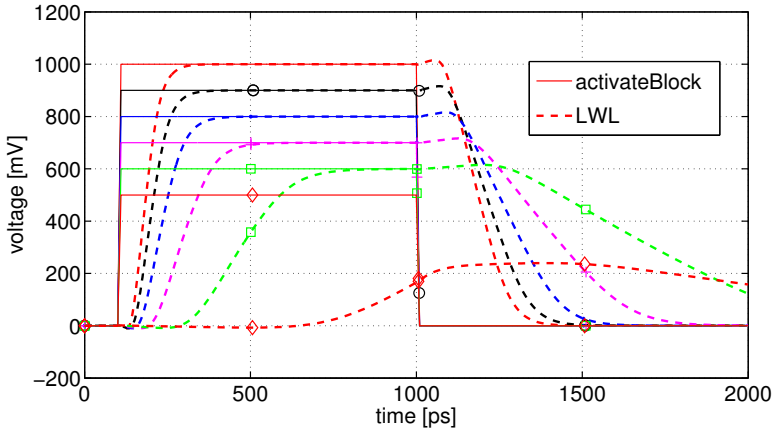
The area of the word line driver is relevant as it is multiplied by the number of words in the memory when fully subdivided word lines are used. If the word line driver is as large as two cells and 32 cells connect to the word line, the area overhead is $2/32 = 6.25\%$.

3.3.1.2 Delay

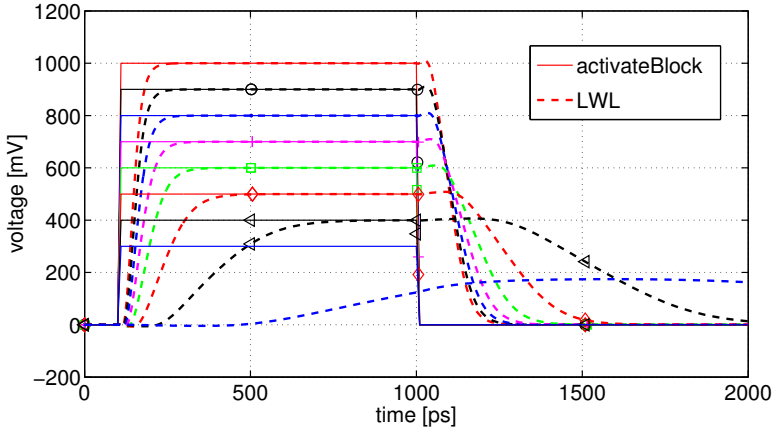
During the read operation, the word line driver is part of the critical path, so the time it takes from a changing input to the point where the word line becomes high is relevant. In our $90nm$ design, we ensured that the global word line GWL has settled before $activateBlock$ is asserted, as it is otherwise very complicated to guarantee the correct order of the different local events. This issue is further addressed in chapter 7. The deactivation delay from the falling edge of $activateBlock$ to the falling LWL must also be reasonably well controlled as the bit line precharge transistors should not be asserted before the word line is disabled. However, this deactivation delay is somewhat less critical. Figure 3.7 shows typical waveforms. Figure 3.8 shows the associated activation delays.

3.3.1.3 Static power consumption

As there are many word line drivers in a memory, their total leakage power can be significant. Figure 3.9 shows the leakage currents for a typical word line driver with SPHVT transistors. The leakage current for this word line driver at $1V$ is $1.8nA$, which corresponds to the leakage of 3 memory cells at $1V$. Figure 3.10 shows how this leakage changes with Vdd, and also shows the leakage currents for a word line driver that uses SPLVT transistors. In the SPLVT word line driver, the leakage current is $110nA$ at $Vdd = 1V$, and this current is strongly dominated by the drain-source current $I_{p,ds}$ of the PMOS transistor in the last invertor. There is also a drain-source leakage path through the NMOS transistors in the NAND gate. Because these NMOS transistors are less wide and stacked, this leakage current is 8 times smaller at $Vdd = 0.5V$ and 30 times smaller at $1V$ than $I_{p,ds}$.



(a) SPHVT transistors



(b) SPLVT transistors

Figure 3.7: Typical waveforms for the word line signal at different supply voltages for the word line driver of figure 3.6.

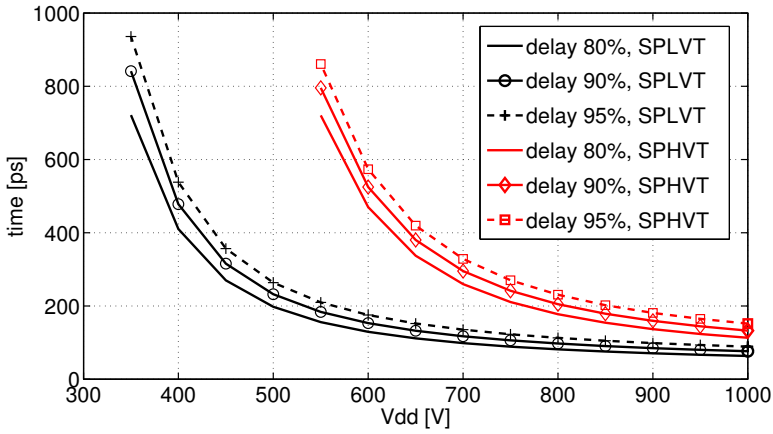


Figure 3.8: The delay of the word line driver from the rising edge of *activateBlock* to different threshold levels of LWL as function of the supply voltage Vdd for the word line driver of figure 3.6.

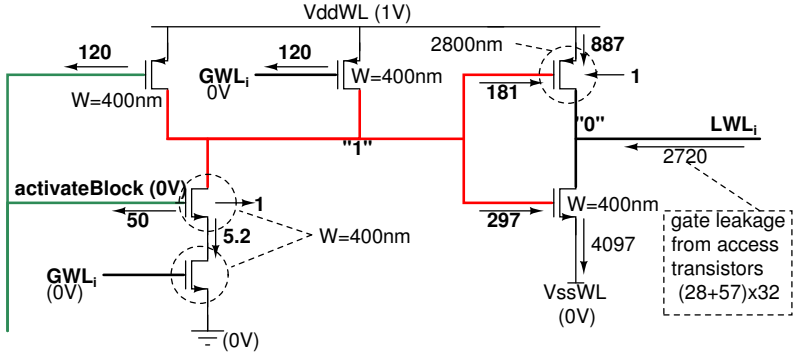
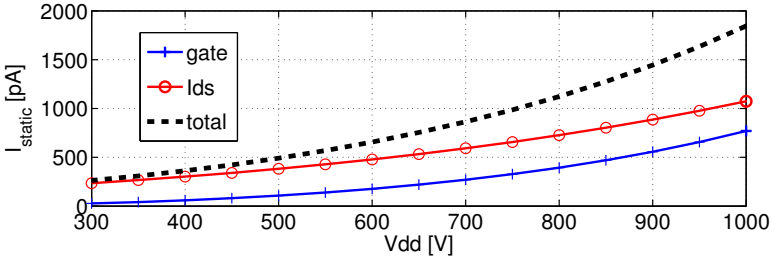
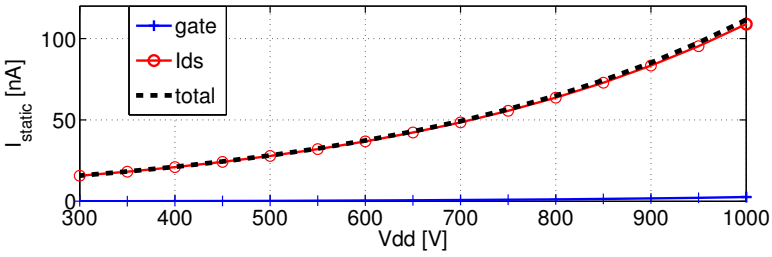


Figure 3.9: Leakage currents for the word line driver in figure 3.6 at $Vdd = 1V$ and with SPHVT transistors.



(a) SPHVT transistors



(b) SPLVT transistors

Figure 3.10: Leakage currents at different supply voltages for the word line driver of figure 3.6.

3.3.1.4 Dynamic energy consumption

Figure 3.11 shows the different contributions to the total energy consumption E_{WL} that is immediately connected to the word line operation. $E_{WL,load}$ is the energy that is used to charge the word line and the gates of the cell access transistors. The word line driver adds two additional energy components. The first one is $E_{WL,driver}$, which corresponds to the charge that is taken from $VddWL$ in figure 3.6, minus the charge that is used for the word line load. In the example, $E_{WL,driver}$ is about 40% of $E_{WL,load}$. This ratio does not significantly change with Vdd. The second additional energy component that is associated with the word line driver, is the energy that is used to drive its input gates. The magnitude of this contribution depends on the number of word line drivers that are attached to each *activateBlock* signal and to each GWL signal, and on the activity of the GWL signal. We will only discuss the contribution of *activateBlock* here, $E_{input,activateBlock}$. If 16 words are connected to each local bit line, it is logical (but not strictly required) to control all 16 word line drivers with a single *activateBlock* signal. In this case, the input gate capacitance of the word line driver is multiplied with a factor of 16, which makes this input gate capacitance a relevant specification of the word line driver. In the example word line driver of figure 3.6, a simulation indicates that the *activateBlock* signal has to deliver $0.88fC$ at $Vdd = 1V$. This contribution is slightly larger than $E_{WL,driver}$ itself, as shown in figure 3.11.

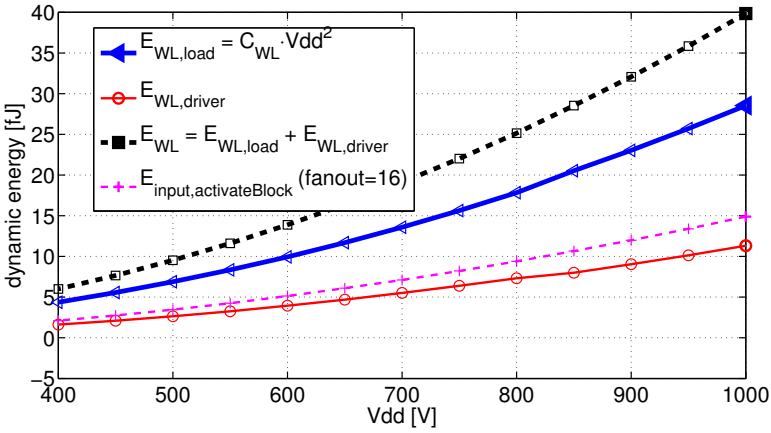


Figure 3.11: The dynamic energy consumption associated with the word line driver from figure 3.6. Plots are for SPLVT transistors, but SPHVT transistors give almost identical results.

3.3.2 Interesting implementations

3.3.2.1 Integrated level conversions

Small voltage level conversions can be integrated efficiently in the word line driver of figure 3.5. For all but one word line driver, LWL is at "0" and $LWLBar$ is at "1". The drain-source leakage of these gates does not increase if $VddWL$ is increased as compared to $VddDecoder$, the supply for GWL and $activateBlock$. $VddWL$ must connect to both the NAND gate and the invertor. The drain-source leakage also does not increase if $VssWL$ for the final invertor is reduced below the ground voltage of the NAND gate.

3.3.2.2 Nand gate with shared transistor

The word line driver in figure 3.12 saves one transistor by sharing the bottom NMOS transistor between the different NAND gates. As only one of the 16 NAND gates can get discharged, the shared transistor can be significantly smaller than the sum of the individual transistors. However, now both $activateBlock$ and $activateBlockBar$ are needed, so two wire capacitances will have to be charged. Notice that $activateBlockBar$ probably has to be pulled up explicitly in between two cycles. Leaving this node floating would be better for static power and dynamic energy, but would result in very dangerous glitches when one of the word lines becomes high while the voltage on the large capacitance is significantly below $Vdd - V_{Tn}$. The small area gain is probably not worth the additional effort.

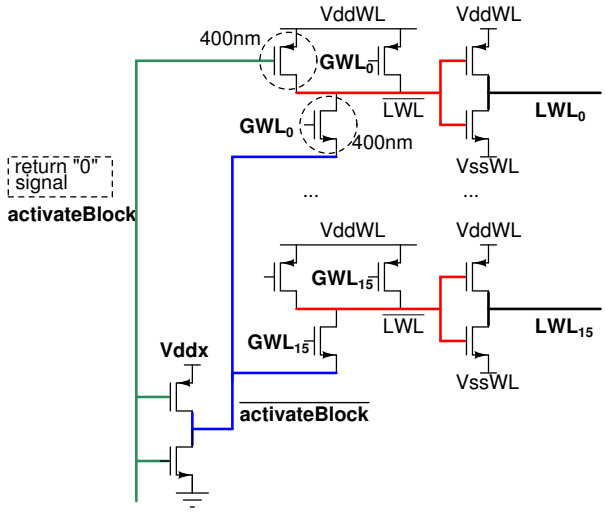


Figure 3.12: Word line driver with a shared NMOS transistor for the NAND gate.

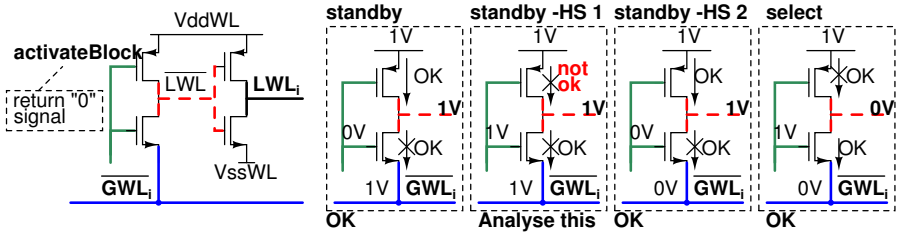


Figure 3.13: Word line driver with a dynamic NAND gate

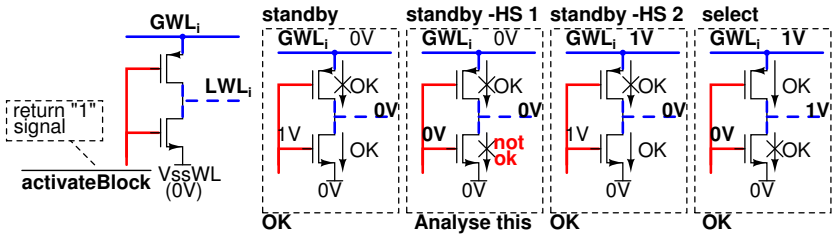


Figure 3.14: Two-transistor word line driver.

3.3.2.3 Dynamic NAND gate with input at source

Figure 3.13 shows the word line driver that was successfully used in our $0.18\mu m$ design (although with other input signals). Because *activateBlock* returns to zero in between every two cycles, it can be used as a precharge signal for a dynamic gate. The NAND-gate is implemented with a single NMOS transistor with *activateBlock* at its gate and $\overline{GWL}$ at its source. This dynamic gate requires only $\overline{GWL}$, not GWL, hence this implementation is much more efficient than that of figure 3.12. In the figure, the different possible combinations of input signals are indicated. In the standby state and in both half select (HS) states, the internal node must stay at "1". The only potential problem occurs in the half select state in which *activateBlock* is asserted. Here, the internal node becomes floating. Although the drain-source leakage can not result in problems, the gate leakage could. Still, in typical technologies and with access times above $10MHz$, this should not be a problem.

3.3.2.4 Voltage amplifying word line drivers

A lot of energy could be saved if the input signals for the word line voltages would not have to be full-swing. In our designs, especially the GWL wire capacitance is significant. In [Mor98, Mai98], a family of half-swing pulse-mode gates was presented, including a three-input AND gate. However, the operational margins of such a gate are not sufficient for use in smaller technology nodes with lower supply voltages. Significantly slimmed down versions of such an approach could still be used with careful analysis. For example, it could be possible to change the voltage levels of $\overline{GWL_i}$ from $0V$ and $1V$ to $0V$ and $0.5V$ in figure 3.13.

3.3.2.5 Two-transistor word line driver

The circuit of figure 3.13 can be further simplified by merging the NAND and buffer stage, as shown in figure 3.14. It has the same well contained problem during the half select situation as 3.13. This circuit has a few interesting properties. First of all, it is small. The drain-source leakage from the pull-up PMOS in the driver is eradicated during standby. The driver for GWL typically must not be changed significantly as it already had to drive the large GWL wire capacitance anyway. The main disadvantage is that in a normal design aimed at high speeds, the PMOS in the word line driver is very large, which would significantly

increase the capacitive load on *activateBlock*. This word line driver however seems to be ideal for slower designs that run at reduced supply voltages. Even with a small transistor width, a low threshold pull-up PMOS provides a higher drive current than a 10 times wider high threshold transistor.

3.3.3 Leakage reduction

Several different approaches have been presented to reduce the leakage current of the word line drivers. As the leakage currents strongly depend on the process technology, different processes will require different techniques.

In [Nii04], the gate leakage of the NMOS in the invertor of the word line driver has a significant impact on the leakage budget. This was solved by providing two NMOS transistors, a small keeper and a big driver transistor. When the word line must be disabled, both NMOS transistors are enabled to ensure fast operation. After a short delay, the gate of the big NMOS transistor is made floating until the next word line activation occurs. As this requires additional transistors which again introduce leakage, this technique is only of use in designs with very large word lines, and only if NMOS gate leakage is clearly dominant. In our example word line driver, this was not the case.

In [Wan09], the drain-source leakage of the PMOS transistor was more important than the NMOS gate leakage. By inserting a stacked PMOS sleep transistor in each word line driver, the total leakage of a 128Kbit sub-array during standby was reduced with 15%, from $12mA$ to $10.2mA$.

3.4 Local control periphery to assist cell operation

As discussed in section 2.5, the operational margins of the 6T memory cell can be extended significantly by applying different bias voltages during read and write operations, and the static current consumption of cells that are not being accessed can be reduced significantly, e.g. by lowering the supply voltage.

In our prototypes, these techniques have not yet been applied. However, future low power designs should certainly implement a leakage reduction scheme. For low voltage operation and for future technology nodes, assists for read and write will be needed.

This section first outlines the options that are available to reduce static power. Then, an overview of techniques to improve read stability, read delay and write-ability is provided.

3.4.1 Static power consumption

The static power consumption of 6T cells that are not being accessed can be reduced by changing the applied voltages. An in-depth treatment of this topic can be found in [Gee09]. The nature of the dominant leakage currents depends strongly on the technology that is used. Section 2.3.1.3 covers the important leakage currents for a SPHVT 6T cell in the example $90nm$ technology.

If the drain-source sub-threshold leakage is dominant, applying reverse back-gate bias can still be very effective in $90nm$ technologies. The sub-threshold leakage through the access transistors can be reduced by applying a super cut-off word line voltage, a technique that was popular in DRAMs. In [Lee08], a word line voltage above Vdd was generated with an on-chip charge pump to put PMOS access transistors in super cut-off.

The bit line leakage can be reduced by making the bit lines floating during non-activity by disabling their precharge circuitry, see e.g. [Lee08, Wan08]. This reduces both the gate leakage and the sub-threshold leakage of the access transistor.

These two techniques can be used without a negative impact on SNM_{hold}. Memory cells that are not being accessed typically have significant slack on SNM_{hold}, which allows for a reduction of the standby supply voltage. For the $90nm$ 6T cell from figure 2.42, the supply voltage can be reduced to $235mV$ before the cell failure rate exceeds 10^{-9}. Figure 2.20 shows that the leakage current for this cell reduces with a factor of 10 when the supply voltage is lowered from $1V$ to $300mV$. Especially gate leakage reduces significantly when the supply voltage is reduced. The voltage can be reduced by decreasing Vdd_{cell} [Gee05, Nii04] or by increasing Vss_{cell} [Zha05]. Typically, the bulk voltages are kept at their original values, as this results in a desirable reverse back-gate bias and reduces the amount of capacitance that must be charged before the cells can be activated again.

Whether it is better to reduce Vdd_{cell} or to increase Vss_{cell} depends on many different factors. A lower Vdd_{cell} puts reverse back-gate bias on the PMOS transistors and decreases the gate leakage of the transistors in the cross-coupled pair and the gate leakage of the access transistor at the cell side that stores a "1". A raised Vss_{cell} puts reverse back-gate bias on the NMOS pull-down transistors. The V_{gs} of the access transistor at the cell side that stored "0" becomes negative, significantly reducing its sub-threshold leakage. Which choice reduces the total leakage the most depends strongly on the technology. In a typical 6T cell, the resistance of Vdd_{cell} is much less important than that of Vss_{cell}, as Vss_{cell} must sink all bit line current during a read operation. For small-grained implementations, increasing Vss_{cell} will hence be more expensive than reducing Vdd_{cell}.

Lowering Vdd_{cell} or increasing Vss_{cell} can be done in a number of ways. A drop of Vdd_{cell} of approximately $M \cdot V_{Tp}$ can be obtained by using a stack of M diode-connected PMOS transistors [Nii04]. Vdd_{cell} can be reduced in a continuous manner by changing the bias voltage on the gate of an NMOS transistor which connects the normal power supply with the virtual power rail of the cells [Gee08]. For both approaches, the value of Vdd_{cell} that is obtained depends strongly on the absolute value of the transistor threshold voltage, which changes from die to die. In [Gee08], this problem was addressed by controlling the bias voltage on the NMOS gate with a feedback system based on a monitor circuit that measures the average value of SNM_{hold} on the actual die with the

actual Vdd_{cell} applied. This not only compensates for changes in V_{Tn} of the bias transistor, but also allows to select the optimal value of Vdd_{cell} for each individual die. Another closed-loop approach was discussed in [Wan07a]. Here, a bank of canary flip-flops (memory cells with deliberately deteriorated data retention voltage) is included. The number of canary flip-flops that fail is used to assess the available margin for the standby power supply. In [Ham09], the problem was addressed with calibration. A programmable voltage generator sets the desired Vdd_{cell}, which is then enforced with a feedback loop including an op-amp. The static power consumption of the op-amp was only 2.5% of the static leakage of the memory array.

Leakage power depends strongly on the available technology options. In [Kim05], higher threshold voltage devices were used. The reduction in performance was countered by the use of forward body bias. This paper also gives a good overview of previous leakage reduction implementations and their impact on performance.

Leakage power can be reduced further by employing a DC-DC convertor to generate the reduced power supplies, as was demonstrated in [Kwo08].

Different choices can be made regarding the control of the cell sleep state. In [Gee05] and [Sal05], a word-by-word scheme was employed. Only the voltage supply of the accessed row is raised as part of the normal memory access. All other cells remain in their standby state. After the access, the accessed cells immediately return to the standby state. This scheme requires that the cell supply voltage is routed along the word line. A similar scheme could be conceived on a local block by local block basis, but this significantly increases the amount of capacitance that is switched each cycle. Such schemes have three main advantages: the control of the standby mode is entirely contained inside the memory block, the amount of capacitance that is switched is always limited and the leakage minimisation is at all times applied to all inactive cells. The main disadvantages are the area overhead and an increased access time.

In [Wan08] on the other hand, explicit software-controlled standby modes are suggested that affect the entire memory at once. Such a scheme introduces less hardware overhead and can probably be implemented without affecting the cycle time if one or a few dedicated wake-up cycles are allowed. The explicit software control might complicate software development. More fundamentally, due to the magnitude of the capacitance that has to be switched when the memory changes between standby and active mode, relatively long periods of inactivity are needed to make the switch to the standby mode worthwhile.

Software control can obviously also be applied with a finer control granularity, e.g. in blocks of 128 words. In these cases, hardware assistance is probably needed to avoid switching blocks on and off too often. The optimal control scheme and control granularity will depend on the amount of data locality in the application domain.

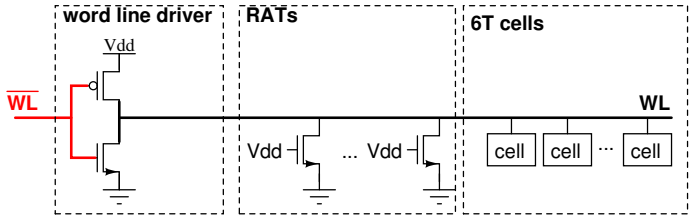


Figure 3.15: Replica Access Transistors (RATs) to reduce the word line voltage in a way that more or less tracks the PVT corners.

3.4.2 Read stability, read current and write-ability

Read stability can be improved by reducing the strength of the access transistors. The main disadvantage of this approach is obviously that this reduces I_{read}. In [Ohb07], the word line voltage during read is reduced by connecting Replica Access Transistors (RATs) to the word line as illustrated in figure 3.15. The RATs are NMOS transistors with their gates connected to Vdd. The word line voltage during read is then determined by the ratio of PMOS to NMOS drive current. This implementation tracks the cell read stability to some extent: with higher V_{Tn} and smaller $Vdd_{cell} = Vdd_{WL}$, the intrinsic cell read stability improves and the impact of the RATs on the effective word line voltage reduces accordingly. Another circuit to reduce the word line voltage is discussed in [Nii08]. In [Hir09], the word line voltage is made programmable between the high cell supply voltage V_{SM} and the lower core supply voltage V_{DD} in 9 steps. The word line voltage is equal for all word line drivers and is selected by fuse programming. Notice that both implementations use the reduced word line voltage not only for the read but also for the write operation. There is a good explanation for this: in memory organisations where multiple words connect to each word line or where write masking is desired, a higher word line voltage for write is simply not possible as it would destabilise the half-selected cells. In such memories, a negative bit line scheme as presented in [Bar08] and [Nii08] could be used to increase the available write margin. This half-select problem is probably the main reason why Vdd_{cell} switching between read and write is receiving more attention in the literature than Vdd_{WL} switching.

Read stability can also be improved by increasing the strength of the pull-down transistor at the cell side that stores a "0". A higher Vdd_{cell} gives the desired result and increases I_{read}. This approach is obviously limited by the maximal allowed voltage stress.

Setting the difference between Vdd_{cell} and Vdd_{WL} allows to trade-off write-ability versus read stability. However, many commercial memories have only two power supplies, Vdd_{memory} which is maintained at its normal high voltage, and the processor core supply, Vdd_{core}, which is subject to dynamic voltage scaling. In such a setup, connecting Vdd_{WL} to Vdd_{core} would result in write-ability issues [Pil08].

The use of a different Vdd_{cell} during read and write operations has attracted quite some attention lately. In [Zha06], [Bar08] and [Pil07], Vdd is routed along the bit lines and is dynamically switched between two externally provided power supplies. The supply voltage of the columns of the half-selected cells is not altered. In [Ohb07], Vdd_{cell} is lowered during write by means of a capacitive control circuit. This results in a floating power net at a reduced voltage during write. Some other memories have been presented in which the power supply is simply made floating, without explicitly reducing its voltage [Sin08a]. Extreme care is needed to ensure correct operation of such schemes under all data patterns.

Read stability can be improved to some extent by reducing the precharge voltage. In [AR08], SNM_{read} improved with +30% by such a scheme for an industrial $45nm$ technology. As shown in figure 2.53, the maximal improvement for our $90nm$ cell is only 11%. Other implementations to obtain a near-optimal precharge voltage use a precharge NMOS transistor, which results in a precharge voltage around $Vdd - V_{Tn}$, e.g. [Bha08].

Read stability can also be improved by raising the trip voltage V_{trip} of the cell invertor from the node that stores "0" to the node that stores "1", as this allows for a larger increase of the voltage on the "0"-node. In [Ham09], this is obtained by applying a forward back-gate bias on the PMOS pull-up transistors during read, which makes the PMOS transistors stronger, as desired. This allowed for a reduction of the active power supply with $70mV$.

Write-ability can be improved significantly by reducing Vdd_{cell} or by increasing Vss_{cell}. A much stronger improvement in WTP is obtained when the change in voltage is applied to only one half of the cell, depending on the data that has to be written. In [Suz08], such a scheme was called differential VSSM biasing (VSSM is Vss_{cell}).

Two comparison papers regarding read-write assists have been published recently, though somewhat limited in scope. In [Bha08], a technical comparison of different schemes is presented. In [Khe08], a high-level overview of different schemes is presented. The authors express a clear preference for read-modify-write schemes, word line lowering for improved read stability and a lowered Vdd_{cell} during write for improved write-ability.

3.5 Bit line periphery and hierarchy

This section discusses the bit line periphery and the bit line hierarchy by means of some interesting example implementations.

3.5.1 Traditional bit line structure

Figure 3.16 shows a rather traditional implementation for the SRAM bit line. The cells connect to a long bit line with a large capacitance. Before the read operation starts, this bit line is precharged high. When the word line is asserted,

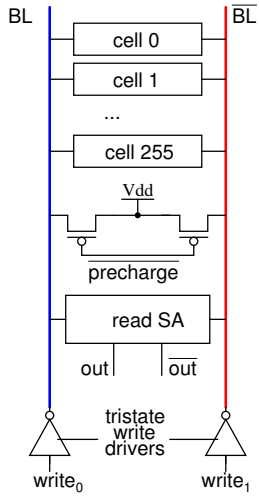


Figure 3.16: The traditional bit line scheme.

the activated cell starts to pull one of its bit lines down. After a certain time $T_{develop}$, the sense amplifier is activated. $T_{develop}$ needs to be large enough so that the developed voltage difference on the bit lines is larger than ΔV_{min}, the smallest input voltage difference that the sense amplifier can sense and amplify reliably. Sense amplifiers are discussed in detail in chapters 4 and 5, but typical values for ΔV_{min} are between $50mV$ and $100mV$ when complementary bit lines are used.

If N cells connect to a bit line, a lower bound for the cell delay and the energy can then be calculated as

$$\begin{aligned} T_{develop} &\geq \frac{C_{BL} \cdot \Delta V_{min}}{I_{read}} \\ &\approx \frac{N \cdot C_{BL,cell} \cdot \Delta V_{min}}{I_{read}} \end{aligned}$$

$$\begin{aligned} E_{BL,read} &\geq C_{BL} \cdot \Delta V_{min} \cdot Vdd \\ &\approx N \cdot C_{BL,cell} \cdot \Delta V_{min} \cdot Vdd \end{aligned}$$

The write energy for this bit line structure is much higher than the read energy, as one of the bit lines needs to be pulled down below the cell's WTP. In practice, it is pulled down to $0V$.

$$E_{BL,write} = C_{BL} \cdot Vdd^2$$

For example, with $N = 256$, $C_{BL,cell} = 0.5fF$, $Vdd = 1V$, $I_{read} = 50\mu A$ and $\Delta V_{min} = 100mV$, $T_{develop} \geq 256ps$, $E_{BL,read} \geq 12.8fJ$ and $E_{BL,write} = 128fJ$.

Obviously, if the SA can be significantly improved, e.g. if ΔV_{min} can be reduced from $100mV$ down to $10mV$, the impact of $C_{BL,cell}$ and I_{read} on $T_{develop}$ and on $E_{BL,read}$ would reduce with a factor of 10. $E_{BL,write}$ would remain equally problematic.

Several variations on this bit line scheme have been used. In figure 23 of [Ito95] for example, bit lines are precharged and clamped to $Vdd - V_{Tn}$ with a diode-connected NMOS. To avoid history effects and the impact of the mismatch in V_{Tn} between the two diode-connected precharge transistors, an additional equalisation transistor is required. A selector that consists of complementary pass transistors connects one of multiple adjacent bit line pairs to the input of the sense amplifier.

In practice, the average bit line swing is always larger than ΔV_{min}. The discharge only stops when the word line is de-activated. With traditional timing, margins have to be included in the timing of this de-activation to guarantee correct operation under all process variations. Replica-based timing has been used to avoid this margin [Amr94]. However, due to the large intra-die variations in recent technologies, different cells on the same die generate different swings in the active time period. As the voltage difference developed by the slowest cell must be larger than ΔV_{min}, the average swing is certainly larger than this value. For a $90nm$ SPHVT cell, the average voltage swing is about 40% larger than the minimal required swing. This is illustrated in section 6.2.3.

An advantage of this bit line structure is that write masking can be implemented in a very simple way: if both bit lines of a bit line pair are kept in their precharged state, the cell is subject to the same conditions as during a read operation and hence maintains its contents.

3.5.2 Domino evaluate read bit line

High-performance memories often use a different bit line structure for read, as shown in figure 3.17. Short single-ended read bit lines are used, most often connecting 8 cells. All bit lines are precharged high. When the cell stores a "0", the local read bit line is discharged entirely. This triggers the NAND gate in the readbuffer. The NMOS driver transistor then discharges GBL entirely, which in turn triggers the next gate. In the figure, this is a 4-input NOR gate with the inputs at the source. Such a ripple evaluation does not require additional timing signals, but the energy consumption is rather high, unless the memory stores much more ones than zeros.

If the previous example with $N = 256$ rows would be implemented with $T = 8$ cells per local bit line, the energy consumption would be

$$\begin{aligned} E_{BL,read} &= p_0 \cdot (E_{RBL} + E_{GBL}) \\ &= p_0 \cdot (T \cdot C_{BL,cell} \cdot Vdd^2 + (N - T) \cdot C_{GBL,wire\,per\,cell} \cdot Vdd^2) \end{aligned}$$

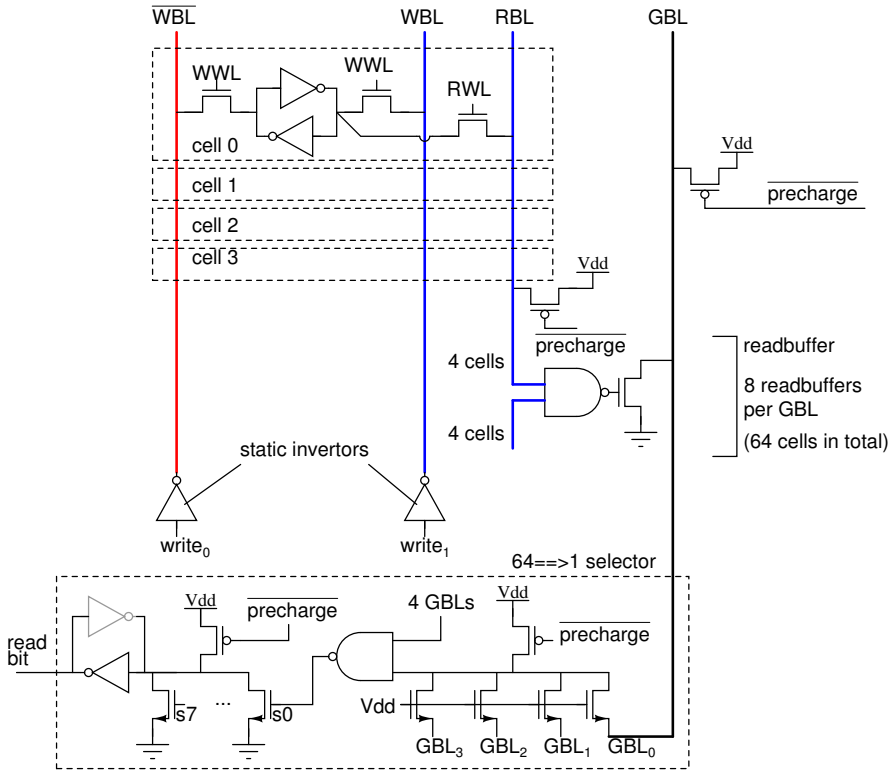


Figure 3.17: Bit line structure with single-ended full-swing dynamic ripple evaluation [Sil98].

Here, p_0 is the probability that a cell stores a "0". With $C_{GBL,wire\,per\,cell} = 0.15fF$ and $p_0 = 0.5$, $E_{BL,read} = 20.6fJ$. There are two reasons why $C_{GBL,wire\,per\,cell}$ is significantly smaller than $C_{BL,cell}$ of a local bit line: fewer gates connect to this wire and therefore, it can be routed in a higher metal layer with a more relaxed wire spacing.

The impact of I_{read} on the memory delay now depends on the trip voltage of the readbuffer, $V_{trip,readbuffer}$:

$$\begin{aligned} T_{develop} &= \frac{C_{RBL} \cdot (V_{precharge} - V_{trip,readbuffer})}{I_{read}} \\ &= \frac{T \cdot C_{BL,cell} \cdot (V_{precharge} - V_{trip,readbuffer})}{I_{read}} \end{aligned}$$

With $T = 8$, $V_{trip,readbuffer} = 0.6V$, $V_{precharge} = 1V$ and $I_{read} = 50\mu A$, $T_{develop} = 32ps$. Obviously, the remaining stages in the evaluation path contribute significantly to the memory delay as well. However, it is clear that the impact of I_{read} itself on the memory delay is very small with this bit line structure.

The implementation in [Sil98] (figure 3.17) contains an additional interesting aspect. A 7T cell is used, with a separate double-ended write port and a single-ended read port. The write bit lines connect to much more cells than the read bit lines, as their delay is less problematic. Separating the two ports has two advantages: the short read bit lines do not need write periphery, and the write bit lines can be controlled with static invertors, which reduces their activity and energy consumption significantly.

3.5.3 Hierarchical divided bit line

In [Kar98], the hierarchical divided bit line structure from figure 3.18 was introduced. Details on the implementation of the peripherals, such as local bit line precharge, were not discussed, and the paper has the look and feel of an unfinished spreadsheet. The bit line is divided in short local bit lines (LBL) that connect through pass transistors to a global bit line (GBL). If the resistance and capacitance of the pass transistor between LBL and GBL could be ignored, the upper bound on performance becomes

$$\begin{aligned} T_{develop} &\geq \frac{(C_{LBL} + C_{GBL}) \cdot \Delta V_{min}}{I_{read}} \\ E_{BL,read} &\geq (C_{LBL} + C_{GBL}) \cdot \Delta V_{min} \cdot Vdd \\ E_{BL,write} &= (C_{LBL} + C_{GBL}) \cdot Vdd^2 \end{aligned}$$

As a first-order estimate,

$$\begin{aligned} C_{LBL} &\approx T \cdot C_{LBL,cell} \\ C_{GBL} &\approx (N - T) \cdot C_{GBL,wire\,per\,cell} \end{aligned}$$

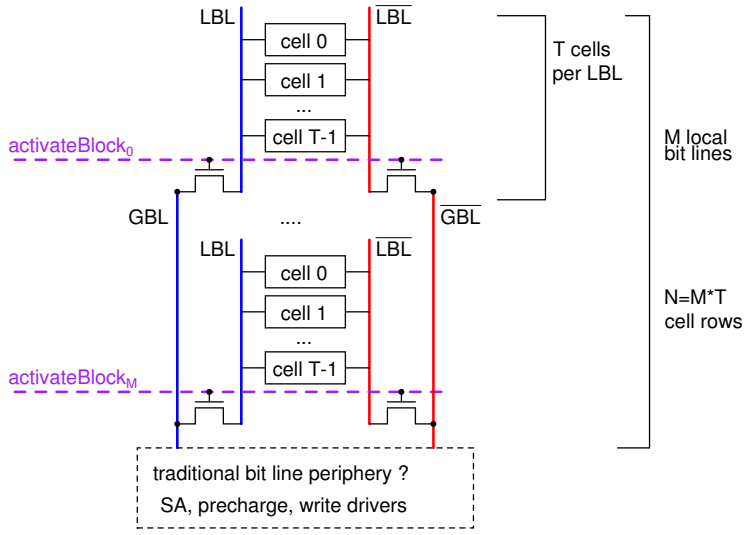


Figure 3.18: Hierarchical divided bit line structure [Kar98].

For example, with $N = 256$, $T = 8$, $C_{BL,cell} = 0.5fF$, $C_{GBL,wire\,per\,cell} = 0.15fF$, $Vdd = 1V$, $I_{read} = 50\mu A$ and $\Delta V_{min} = 100mV$, $T_{develop} \geq 82.4ps$, $E_{BL,read} \geq 4.12fJ$ and $E_{BL,write} = 41.2fJ$. All these specs are considerably better than for the traditional setup. In practice, the pass transistors have a profound impact on the resistance and, and the additional bit line periphery increases the bit line capacitance somewhat. The additional pass gates need control signals, which would increase $E_{BL,read}$ in our example by about $1fJ$. Also, as more peripheral circuits are added, the height of the matrix increases, which in turn increases C_{GBL}. Some of these effects were discussed in [Kar98], but the underlying implementation assumptions were not adequately explained. Even with these undesired effects added to the picture, a subdivided bit line scheme still reduces the energy consumption. The impact on write energy is certainly substantial.

In [Kar98], a slow yet interesting method to control the swing on the GBL was discussed. First, both LBL and GBL are precharged. Then, the word line is activated while the pass transistor between LBL and GBL is disabled. The LBL is discharged to ground, and the word line is disabled again. When the pass transistor between LBL and GBL is activated, capacitive charge distribution between LBL and GBL generates a well controlled small voltage swing on GBL.

In [Amr00], subdivided bit lines were included in an extensive and interesting analysis of speed and power scaling of memories. However, no pointers to practical implementations were provided.

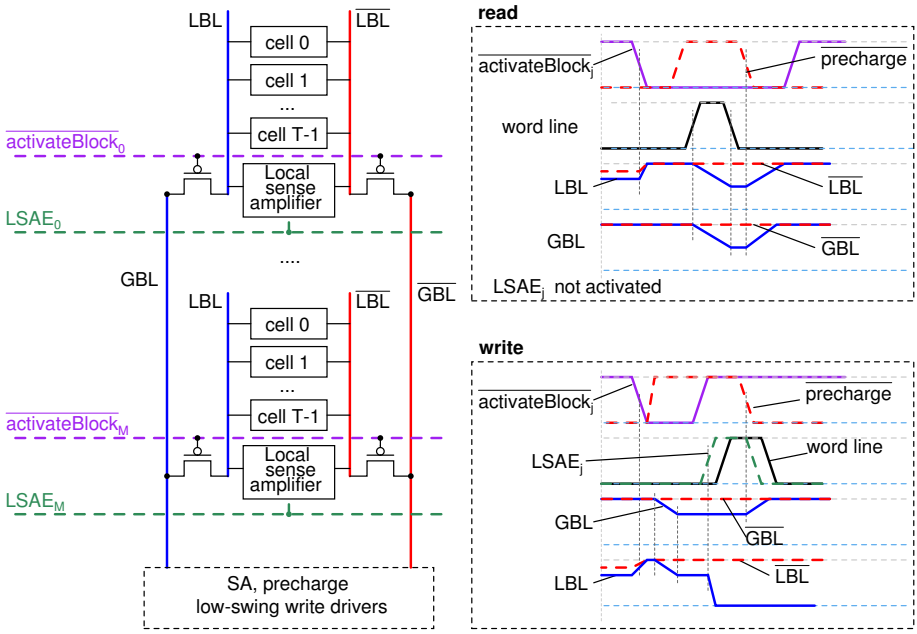


Figure 3.19: Hierarchical bit line structure with local sense amplifier [Yan05].

3.5.4 Hierarchical bit line with local SA

In [Yan05], the very interesting practical implementation for subdivided bit lines from figure 3.19 was proposed. By adding a local sense amplifier to each local bit line pair, a low-swing signal can be used on the global bit lines during write, which reduces the write energy. This is also the first paper on subdivided bit lines that provides an adequate discussion of the circuit-level operation.

The pass transistors between GBL and LBL are implemented with PMOS transistors. The precharge of the local bit lines is performed through these pass transistors. During read, the pass transistors are enabled and both GBL and LBL are precharged high. The word line is activated, and a voltage difference is developed on both LBL and GBL. As in the traditional scheme, a global read sense amplifier senses this difference. The local sense amplifier is not enabled during the read operation. During write, a small voltage difference is created on the GBLs according to the data to be written. This voltage difference is transfered to the LBLs through the PMOS access transistors. When the PMOS access transistors are disabled, the local sense amplifier is enabled, amplifying the data to be written on the local bit lines to a full-swing signal. Finally, the word line is asserted and the cell gets written.

The local sense amplifier acts as a low-swing write receiver that enables a reduced swing $\Delta V_{min,LSA}$ on the global bit lines during write. This reduces the

lower bound on the write energy to

$$E_{BL,write} \geq (C_{LBL} \cdot Vdd + C_{GBL} \cdot \Delta V_{min,LSA}) \cdot Vdd$$

For example, with $N = 256$, $T = 16$, $C_{BL,cell} = 0.5fF$, $C_{GBL,wire\,per\,cell} = 0.15fF$, $Vdd = 1V$ and $\Delta V_{min,LSA} = 200mV$, $E_{BL,write} = 15.2fJ$. $E_{BL,write}$ consists of $8fJ$ LBL energy and $7.2fJ$ GBL energy. Even if the local sense amplifier adds $4fJ$ to the write energy budget, this low-swing write scheme halves the write energy as compared to a normal subdivided bit line scheme as in [Kar98].

3.5.5 Per-bit SA scheme

In [Tak04], a per-bit sense amplifier scheme was introduced. The authors observed that performing column selection with pass transistors placed before the SA scales poorly, both to smaller technology nodes and to shorter bit line lengths. Their solution is shown in figure 3.20. The SA is moved in front of the selector, which can then be implemented as a dynamic NAND gate with full-swing inputs. The specific implementation of the structure assumes that the word line connects to (at least) 8 different words, with their cells interleaved. The content of only one out of eight adjacent cells is actually desired.

The local bit lines, the global bit lines and the write data lines are all full-swing. Although the explanation of the authors starts from the traditional organisation, as discussed in section 3.5.1, it is at least as logical to interpret this implementation as a variation on the domino ripple evaluation as discussed in section 3.5.2, in which a local sense amplifier is added. The local sense amplifier is used only during read, and can be triggered as soon as the cell has developed a large enough swing on the local bit lines, e.g. $\Delta V_{min,LSA} = 200mV$. Once the sense amplifier is activated, it speeds up the signal evolution on the local bit line. This allows to reduce the memory delay or to attach more cells to a local bit line for the same memory delay.

This scheme further reduces the impact of the cell read current I_{read} as compared to the domino ripple evaluation. The cell must now only sink a current $Q_{cell} = C_{LBL} \cdot \Delta V_{min,LSA}$ rather than $Q_{cell} = C_{LBL} \cdot (V_{precharge} - V_{trip,readBuffer})$. From that point on, the local sense amplifier takes over.

In [Tak06], the same author proposes a new cell type, the read-static-noise-margin-free 7T cell from figure 2.63. Figure 3.21 shows the local bit line periphery that was used in this design. A single-ended local read bit line with 8 cells is used. The write bit lines are not subdivided. Precharge, write and read buffer are implemented with just 6 transistors. The PMOS keeper transistor slows down the read operation, and at first sight is not needed for read. However, the memory supports low voltage operation, and this might introduce signal integrity issues on the floating bit line. During write, the PMOS keeper could prove useful to write "1" to the cell, although this should not be a real issue with this 7T cell.

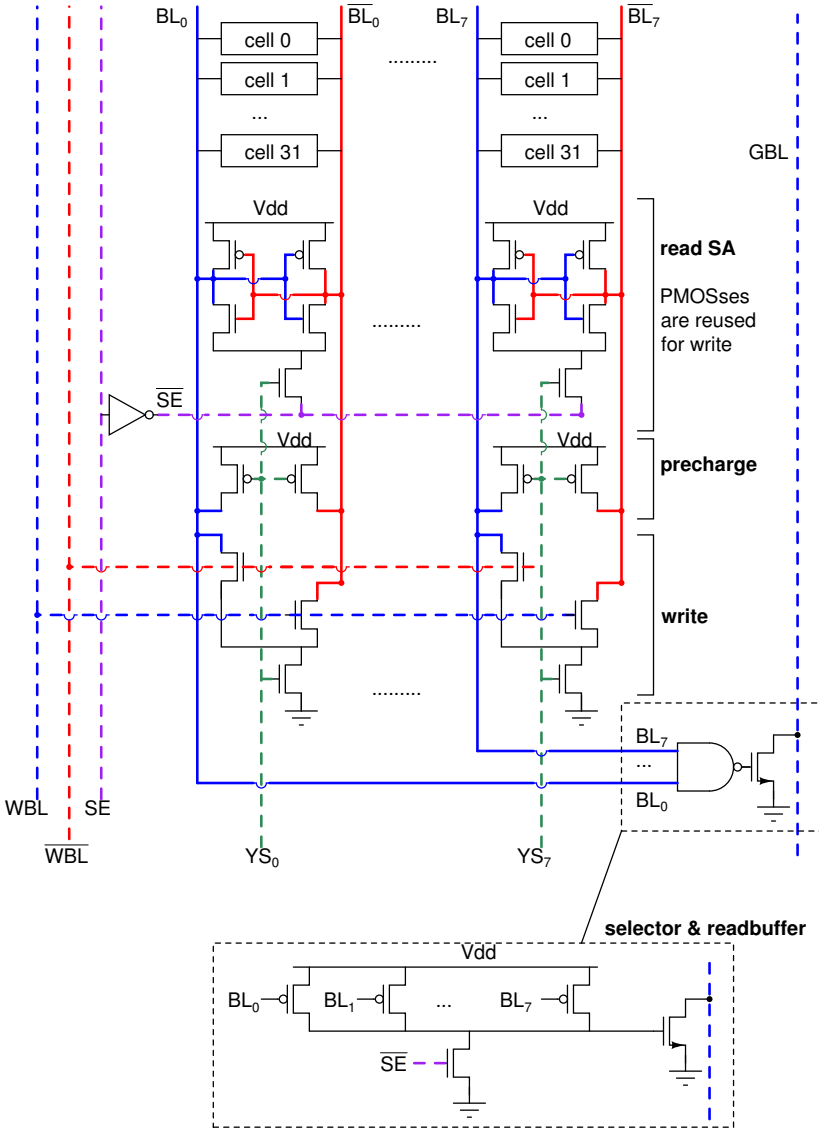


Figure 3.20: Per-bit SA scheme as used in [Tak04].

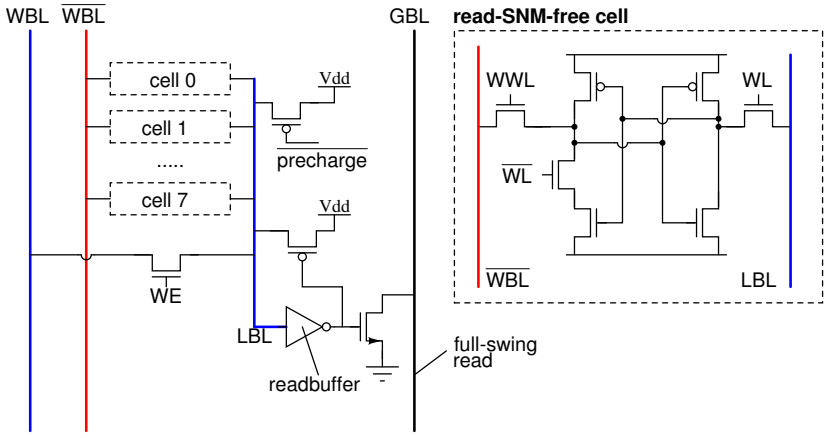


Figure 3.21: The local bit line periphery as discussed in [Tak06].

Because the global lines for read and write are again separated, static write drivers could be used, but this was not discussed. With this cell and bit line structure, it seems possible to use reduced-swing write bit lines, e.g. from $0V$ to $Vdd/2$.

It might be interesting to note that the authors explicitly claim that the fully subdivided word line scheme which they used in this design is superior over the interleaved design they used in their previous work. The area overhead of fully subdivided word lines was estimated at 10% for a word length of 16 bits.

3.5.6 Short buffered bit line and shared write receiver

In our first design, in a $0.18\mu m$ technology, the local bit line periphery from figure 3.22 was used [Cos07]. Short buffered bit lines reduce the importance of the cell current I_{read} as explained in section 3.5.2. This implementation contains two interesting new ideas.

Firstly, this is the first design in which a readbuffer is used in combination with a low-swing GBL read scheme. At the beginning of a read cycle, the GBL is precharged to a low voltage, e.g. $200mV$. This way, the voltage swing on the GBL is well controlled without having to rely on accurate timing control. When the low precharge voltage is supplied separately, or generated with a DC-DC converter, an even larger reduction of the energy consumption could be obtained. This has not been addressed in our work though. More information on this topic is provided in chapter 6.

Secondly, by sharing the low-swing write receivers between several local bit lines, most of the energy reduction of the local sense amplifier scheme from section 3.5.4 is obtained at a lower area cost.

This implementation of the local periphery contained some additional techniques which were very useful in a $0.18\mu m$ technology, but which can not be transferred

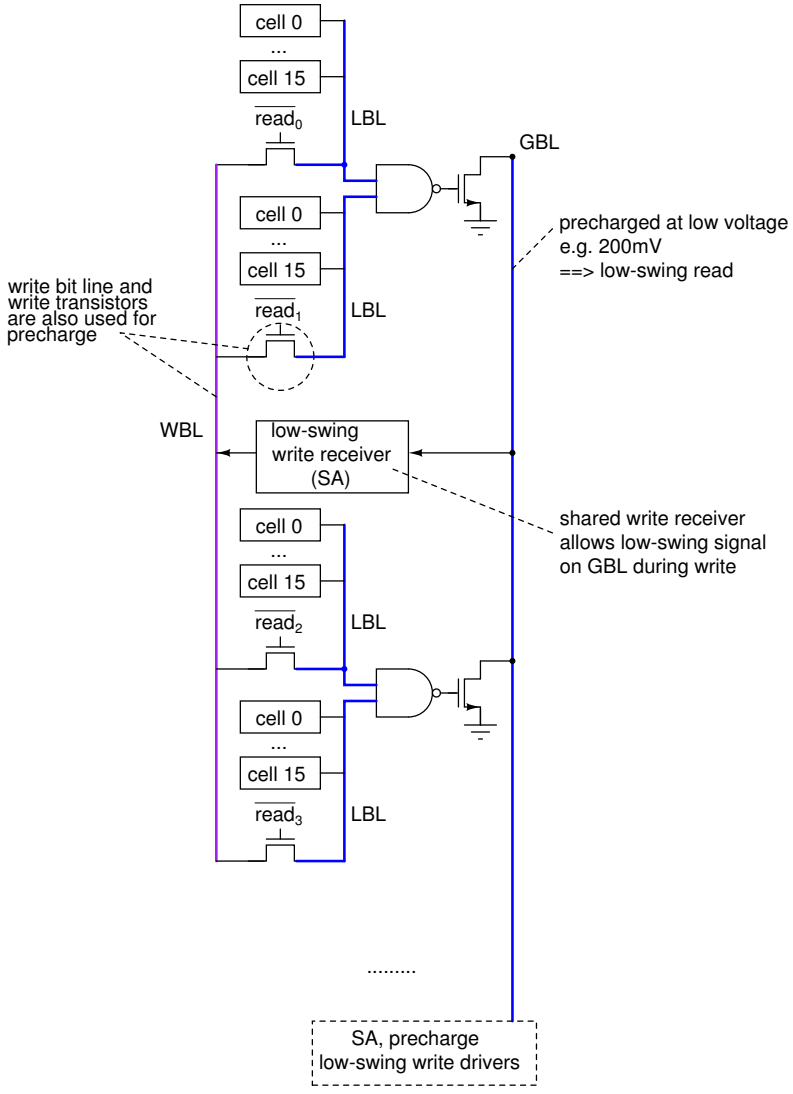


Figure 3.22: A short buffered bit line with shared write receiver [Cos07].

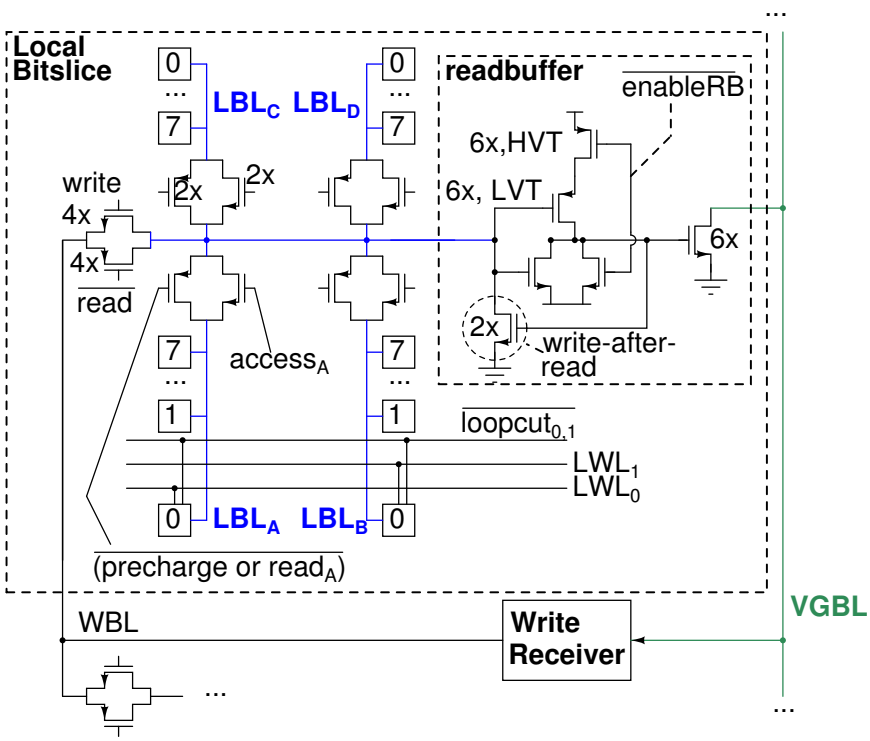


Figure 3.23: The local bit slice from [Cos09].

to $90nm$ and beyond. The local bit line precharge voltage was reduced from $1.6V$ to $1V$, which significantly reduces the energy consumption associated with the full-swing local bit line. This lower LBL voltage also allows to implement both LBL precharge and LBL write with a single NMOS transistor.

This short buffered bit line also allows the exploitation of dynamic read stability, which is discussed in section 3.6.

3.5.7 Short buffered bit line with write-after-read

Our second design, in a $90nm$ technology, used the local bit line periphery as shown in figure 3.23 [Cos09]. There are two interesting things that should be noted here. First, the readbuffer is equipped with a write-after-read transistor to stabilize the most unstable memory cells during read operations. Second, the readbuffer has been extended to a NAND gate with an explicit control signal that arrives at a high threshold PMOS transistor. This allows the use of a low threshold PMOS in the readbuffer so that it is triggered at a higher LBL voltage. The explicit reset of the readbuffer also resolves the latching risk that was introduced by the read-after-write NMOS transistor. This however does impact the dynamic energy consumption.

Performing the selection before the readbuffer seems to be a less interesting idea, as it requires complementary pass transistors and rather involved control signals.

3.5.8 Overview

Table 3.1 provides a very idealised overview of the different options for local bit line periphery and hierarchy. A more in-depth treatment of the performance of data transfer schemes is provided in chapter 6.

$T_{develop}$ is a good measure of how important the cell read current is in the total memory delay. Smaller values reduce the impact of a small I_{read}. This is important when transistors with high threshold voltages are used in the cell to reduce leakage, or when the cell current suffers badly from intra-die variations. For this metric, the per-bit SA scheme (E) that uses a local read sense amplifier performs best for a given value of T. Schemes with short buffered bit lines (B,F) perform second best, followed by schemes with sub-divided bit lines (C,D). The traditional scheme (A) performs the worst.

The bit line read energy is lowest for the pass transistor based subdivided bit line schemes (C,D), followed by the short buffered bit line with low-swing global bit line (F). The traditional implementation (A) consumes less energy than the two options that use full-swing single-ended global bit lines (B,E).

The bit line write energy is lowest for the schemes that use local low-swing write receivers (D,F). A scheme with static global write bit lines (E) is second best. A complementary full-swing signal on the precharged global bit lines is costly (C), but not as costly as full-swing signals on a write bit line that connects to each cell (B), even if these signals are static. If the signals on the bit lines are precharged each cycle, the energy consumption is twice as high (A).

A. Traditional bit line structure		
$T_{develop}$	$\frac{N \cdot C_{BL,cell} \cdot \Delta V_{min}}{I_{read}}$	$256ps$
$E_{BL,read}$	$N \cdot C_{BL,cell} \cdot \Delta V_{min} \cdot Vdd$	$12.8fJ$
$E_{BL,write}$	$N \cdot C_{BL,cell} \cdot Vdd^2$	$128fJ$
B. Domino evaluate read bit line		
$T_{develop}$	$\frac{C_{LBL} \cdot (V_{precharge} - V_{trip,readbuffer})}{I_{read}}$	$32ps$
$E_{BL,read}$	$p_0 \cdot (C_{LBL} \cdot Vdd^2 + C_{GBL} \cdot Vdd^2)$	$20.6fJ$
$E_{BL,write}$	$2 \cdot p_0 \cdot (1 - p_0) \cdot C_{WLBL} \cdot Vdd^2$	$64fJ$
C. Hierarchical divided bit line		
$T_{develop}$	$\frac{(C_{LBL} + C_{GBL}) \cdot \Delta V_{min}}{I_{read}}$	$82ps$
$E_{BL,read}$	$(C_{LBL} + C_{GBL}) \cdot \Delta V_{min} \cdot Vdd$	$4.1fJ$
$E_{BL,write}$	$(C_{LBL} + C_{GBL}) \cdot Vdd^2$	$41fJ$
D. Hierarchical bit line with local SA		
$T_{develop}$	$\frac{(C_{LBL} + C_{GBL}) \cdot \Delta V_{min}}{I_{read}}$	$82ps$
$E_{BL,read}$	$(C_{LBL} + C_{GBL}) \cdot \Delta V_{min} \cdot Vdd)$	$4.1fJ$
$E_{BL,write}$	$(C_{LBL} \cdot Vdd + C_{GBL} \cdot \Delta V_{min,LSA}) \cdot Vdd$	$11.4fJ$
E. Per-bit SA scheme		
$T_{develop}$	$\frac{C_{LBL} \cdot \Delta V_{min,LSA}}{I_{read}}$	$16ps$
$E_{BL,read}$	$C_{LBL} \cdot Vdd^2 + p_0 \cdot C_{GBL} \cdot Vdd^2$	$22.6fJ$
$E_{BL,write}$	$(2 \cdot p_0 \cdot (1 - p_0) \cdot C_{WGBL} + C_{LBL}) \cdot Vdd^2$	$22.6fJ$
F. Short buffered bit line and shared write receiver		
$T_{develop}$	$\frac{C_{LBL} \cdot (V_{precharge} - V_{trip,readbuffer})}{I_{read}}$	$32ps$
$E_{BL,read}$	$(C_{LBL} \cdot Vdd + C_{GBL} \cdot \Delta V_{min}) \cdot Vdd$	$7.7fJ$
$E_{BL,write}$	$(C_{LBL} \cdot Vdd + C_{GBL} \cdot \Delta V_{min,LSA}) \cdot Vdd$	$> 11.4fJ$

Table 3.1: Overview table for bit line periphery and hierarchy. $C_{LBL} = C_{WLBL} = T \cdot C_{LBL,cell}$ and $C_{GBL} = C_{WGBL} = (N - T) \cdot C_{GBL,cell}$. Example values are idealised values for $N = 256$, $T = 8$, $C_{BL,cell} = C_{LBL,cell} = 0.5fF$, $C_{GBL,cell} = 0.15fF$, $I_{read} = 50\mu A$, $\Delta V_{min} = 100mV$, $\Delta V_{min,LSA} = 200mV$, $V_{precharge} = 1V$, $V_{trip,readbuffer} = 0.6V$, $p_0 = 50\%$ and $Vdd = 1V$.

3.6 Dynamic read stability

3.6.1 Concept and proliferation

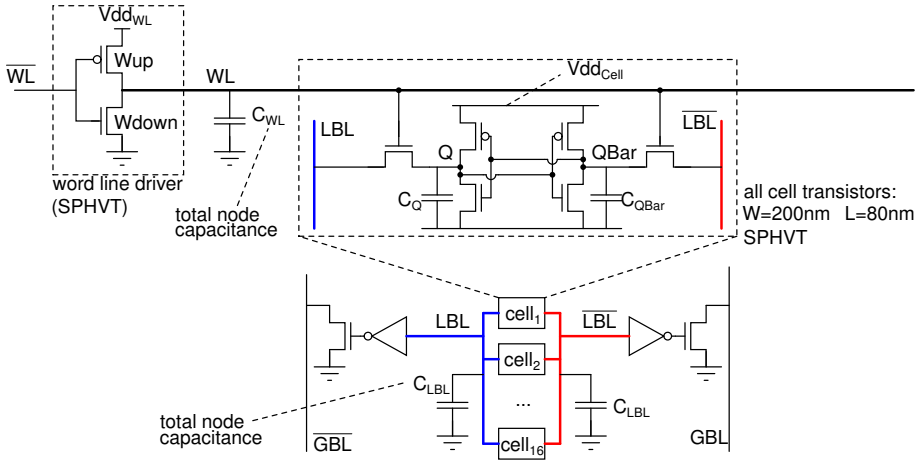
The static noise margin SNM_{read} is the traditional metric to quantify cell stability during the read operation. However, when short buffered bit lines as in section 3.5.6 are used, this metric is much too pessimistic, as can be seen from figure 3.24. Even though the cell has a negative SNM_{read} when operated at $Vdd_{WL} = 1V$ and $Vdd_{cell} = 0.7V$, the cell does not suffer from read upset. This can be easily understood: in the beginning of the read operation, the bit line voltage is close to $V_{BL,precharge}$, but the word line is still significantly below its final value Vdd_{WL}. By the time the word line voltage approaches Vdd_{WL}, the bit line is already discharged significantly, hence reducing the stress on the cell. The values that are used for the total node capacitances in this example are representative for a $90nm$ technology with 16 cells per local bit line and 32 cells per word line. Typically, the capacitance at the internal cell nodes is dominated by the drain and gate capacitances of the cell transistors. This capacitance (approximately $1fF$ in $90nm$ technology with $W = 200nm$) is obviously included in all simulations. C_Q indicates the additional capacitance on the node, e.g. due to wire capacitances, and is typically not significant.

In [Cos06, Cos07], we explicitly discussed the observation that SNM_{read} is too pessimistic when short buffered bit lines are used. We introduced the concept of dynamic stability and showed that this could be used to our advantage by designing a fully functional $5T$ cell with negative SNM_{read}. For the analysis of this cell, we introduced the read margin ΔV_{min} as the minimal difference between the cell nodes, as indicated on figure 3.24c. We used extensive Monte Carlo simulations in all process corners to verify the correct operation. Most of these findings were rediscovered in [Yam08].

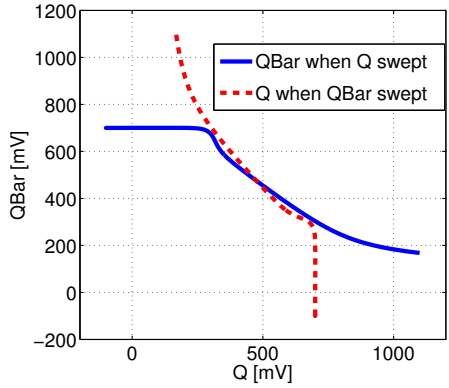
Several other recent papers discuss similar observations. In [Khe06], a pulsed word line scheme is employed, and the impact of the word line pulse duration on cell failure rate is measured. As they use 128 cells per bit line, the reported improvement in read failure rate with a factor 30 is mostly due to the fact that the word line voltage does not reach Vdd_{WL}. Notice that a factor 30 difference in failure rate is for example obtained when all cells up to $\mu + 6.2\sigma$ operate correctly instead of all cells up to $\mu + 5.6\sigma$ (when the distribution is Gaussian). Hence, the region of correct operation is extended by about 0.6σ. To enable the use of a longer word line pulse during write than during read, a read-modify-write scheme must be adopted in this design because the word line connects to multiple words. The paper also describes a pulsed bit line scheme, but this seems a bit too involved as compared to other techniques to reduce $V_{BL,precharge}$.

In [Khe07], a dynamic estimation approach for stability is discussed as a tool to study the effect of supply noise on SRAM stability. Their dynamic stability metric is a binary succeed/fail that is obtained from the transient simulation.

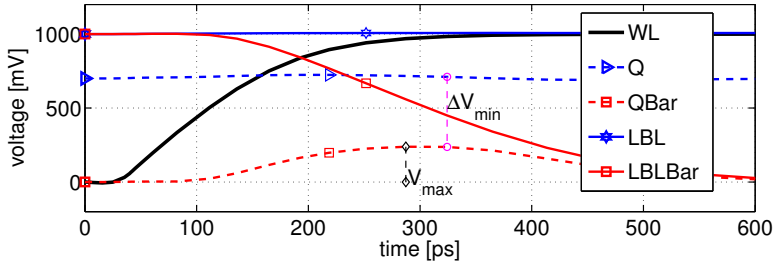
In [Pil07], an interesting read-assist feature is presented. The SAs are moved in front of the column selector, so they connect directly to the local bit lines, which



(a) Circuit setup



(b) Butterfly curves for SNM_{read}



(c) Waveforms for the read operation

Figure 3.24: With short buffered bit lines, a cell can have negative SNM_{read} without suffering from read upset. ($C_{LBL} = 10fF$, $C_Q = 0$, $Vdd_{WL} = 1V$, $Vdd_{cell} = 700mV$, $V_{BL,precharge} = 1V$, $Wup = 1000nm$, $Wdown = 400nm$, $C_{WL} = 24fF$)

connect to 64 cells. The SA can be used to speed up the signal development on the bit lines both for cells during read and for half-selected cells during write. This reduces the number of read upsets with a factor of 5 to 10. In this scheme, all activated local bit lines perform a full-swing transition, so the energy consumption increases significantly. To counter this, the read-assist is only used in the sub-arrays that contain cells that would fail otherwise. Settings are obtained once by calibration after fabrication and written to electronic fuses. This read-assist technique can be considered as a write-after-read scheme to enhance dynamic stability. A similar write-after-read scheme is used in combination with short local bit lines in [Kus09].

In DRAMs, dynamic stability with write-after-read has always been the basic operation mode. Some SRAM cells have been equipped with additional capacitors to improve their soft error rate [Sat99, Jun03, Gep04]. Such added capacitors have not yet been used to improve the cell's read stability though.

In [Dor07], a domino read structure is used with 16 cells per local bit line. The authors mention that short bit lines improve cell stability, but no analysis or numbers are provided. In [Kaw08b], fine-grained bit line segmentation with 16 cells per buffered local bit line is used to improve cell read stability. Finally, [Sha09] provides a discussion on dynamic cell stability based on the state space analysis of the SRAM cell. It is an interesting discussion, but the proposed dynamic criteria for data stability are too involved for practical use.

3.6.2 Metrics for dynamic read stability

For static read stability, there is a generally accepted stability metric, SNM_{read}, that changes smoothly with variations in transistor parameters. A similarly well behaved metric for dynamic read stability would be very useful. Figure 3.24c shows the only two usable metrics that have been used up to now as metrics for the dynamic read stability. ΔV_{min} is the minimal voltage difference during the access between the two cell nodes. As long as the voltages on node Q and $QBar$ do not cross, the cell data is retained. An alternative measure is V_{max}, the maximal voltage on the cell node that stores a "0". As long as the voltage on this node remains below the trip point of the cell's invertor that is controlled by this voltage, data will be retained.

Figure 3.25 shows the circuit setup that will be used in this section to assess the impact of mismatch in transistor parameters on the proposed margins for dynamic read stability. With $Q =$ "1", a positive V_n reduces the drive strength of the critical pull-down transistor. Figure 3.26 shows how V_{max} and ΔV_{min} degrade as the mismatch increases. Curves for different values of C_{LBL} are shown.

These graphs show two important things. First, both V_{max} and ΔV_{min} suffer from a rather abrupt change in the vicinity of the failure point. Any assumption of a Gaussian distribution for these metrics must be rejected. Second, this graph gives a clear indication of the difference between failure predictions calcu-

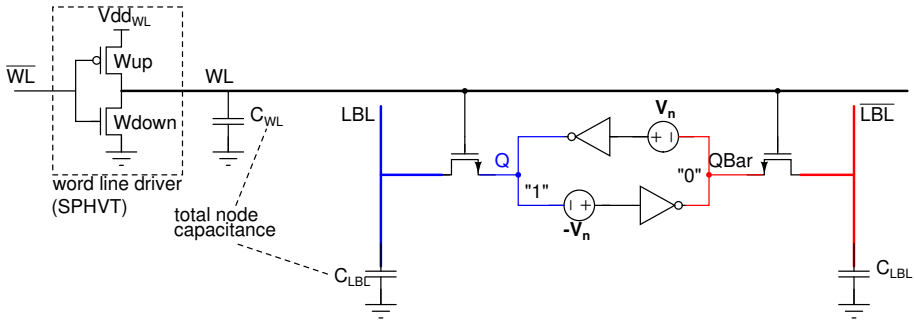


Figure 3.25: Circuit setup including noise sources V_n, which mimic transistor mismatch in the cross-coupled pair.

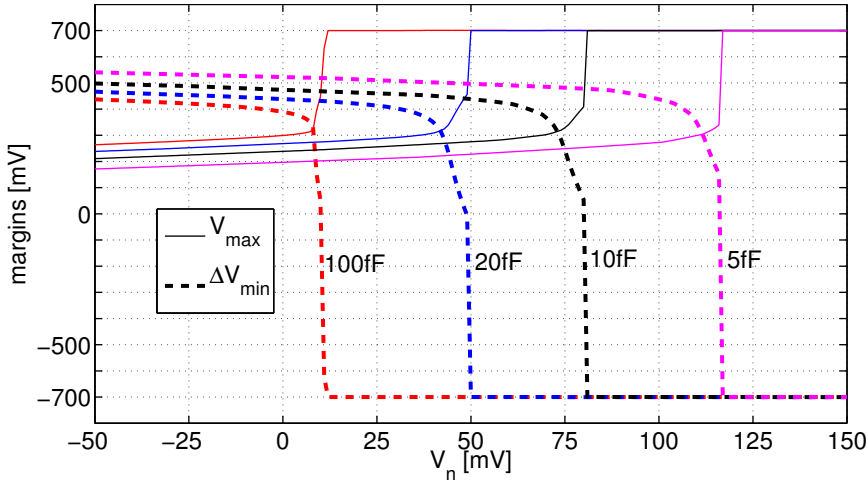


Figure 3.26: Dynamic stability metrics V_{max} and ΔV_{min} for the circuit setup of figure 3.25 as function of V_n, and for different values of C_{LBL} as indicated on the curves. ($Vdd_{WL} = 1V$, $Vdd_{cell} = 700mV$, $V_{BL,precharge} = 1V$, $Wup = 1000nm$, $Wdown = 400nm$, $C_{WL} = 24fF$)

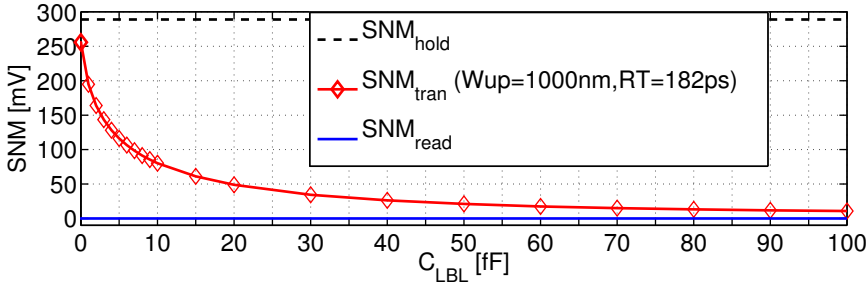


Figure 3.27: SNM_{tran} as function of C_{LBL}. Wup is the width of the PMOS transistor in the word line driver, RT is the rise time of the word line from 10% to 90%. ($Vdd_{WL} = 1V$, $Vdd_{cell} = 700mV$, $V_{BL,precharge} = 1V$, $Wup = 1000nm$, $Wdown = 400nm$, $C_{WL} = 24fF$)

lated with SNM_{read} and the real failure rate as shown by the dynamic stability metrics. With very large C_{LBL} and a long word line pulse, the failure points obtained by SNM_{read} and by transient analysis must be equal. However, even with $C_{LBL} = 100fF$ and a word line pulse duration of $1ns$, the transient simulation only fails when $V_n = 9mV$, while SNM_{read} was 0 at $V_n = 0$. When C_{LBL} is reduced to $5fF$, the cell only fails at $V_n = 110mV$. A cell connected to a short bit line ($5fF$) can tolerate a $101mV$ larger V_n than a cell connected to a long bit line ($100fF$).

As V_n was also the noise source that is used in the definition of SNM_{read} and SNM_{hold}, it is reasonable to use it in the definition of a new metric for dynamic read stability as well. Hence, we define SNM_{tran}, the static noise margin during a transient read operation as the smallest value of $|V_n|$ that causes a read upset in the actual transient read operation. Notice that the static in SNM_{tran} is not a misnomer, as the available noise margin is indeed expressed in terms of a static voltage source V_n.

In the previous example, SNM_{tran} is $9mV$ larger than SNM_{read} when C_{LBL} is $100fF$, and SNM_{tran} is $110mV$ larger than SNM_{read} when $C_{LBL} = 5fF$. In the limiting case, e.g. with a slower word line rise time, SNM_{tran} becomes equal to SNM_{hold}. If the word line rise time becomes too slow, the impact of the leakage currents of the other cells must be investigated. By moving to a short local bit line ($5fF$), a $110mV$ larger V_n can be tolerated. This value can immediately be compared to SNM_{read}. In section 2.5, a $90nm$ cell with all SPHVT transistors with $200nm$ width was discussed. At $Vdd_{cell} = Vdd_{WL} = 750mV$, its SNM_{read} had a mean value $\mu = 142mV$ and a standard deviation $\sigma = 21mV$. From these numbers, it is obvious that $110mV$ additional margin against mismatch is a significant improvement.

Figure 3.27 clearly shows how SNM_{tran} evolves from a value close to SNM_{read} for large values of C_{LBL} to a value closer to SNM_{hold} for very small values of C_{LBL}. This graph is for $Vdd_{Cell} = 700mV$ and $Vdd_{WL} = 1000mV$. Figure

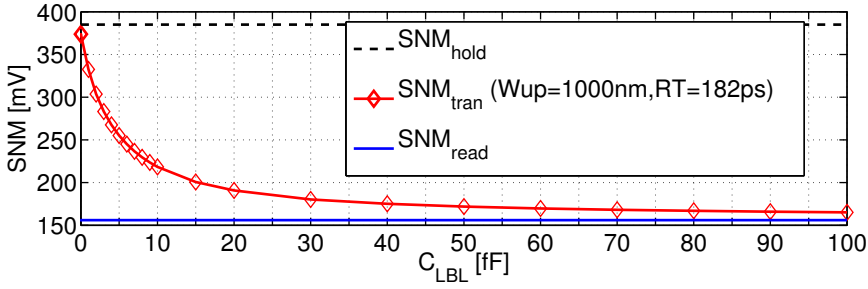


Figure 3.28: SNM_{tran} as function of C_{LBL}. ($Vdd_{WL} = 1V$, $Vdd_{cell} = 1V$, $V_{BL,precharge} = 1V$, $Wup = 1000nm$, $Wdown = 400nm$, $C_{WL} = 24fF$)

3.28 shows the same curve, but at $Vdd_{Cell} = Vdd_{WL} = 1V$. As the cell now sinks more charge from the bit line during the time period that the word line voltage has not reached its peak voltage, SNM_{tran} gets closer to SNM_{hold}.

Figure 3.29 shows empirical CDFs for the different cell metrics for two situations: $Vdd_{cell} = 700mV$ and $Vdd_{cell} = 600mV$. In both cases Vdd_{WL} is $1V$. It is clear that ΔV_{diff} does not follow a Gaussian distribution. For the current settings, V_{max} still seems rather well behaved. However, with a slightly worse cell, sudden failure would occur here as well. SNM_{tran} behaves pretty much as expected, and can be adequately approximated with a Gaussian distribution.

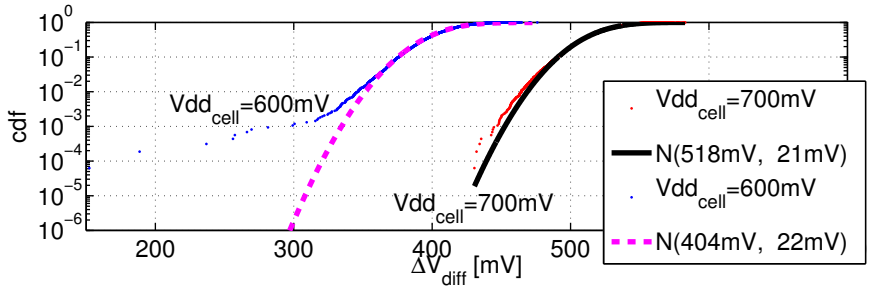
The CDF of the delay (figure 3.29d) highlights the fact that cell stability is only half of the problem. The memory access also needs to finish in time, and this requires a separate analysis, which is not provided here. If the SNM_{tran} analysis is performed, a simple delay analysis based on a response surface that includes only the access transistor and the pull-down transistor of the cell side that stores a "0" should be sufficient.

3.6.3 Design options to improve SNM_{tran}

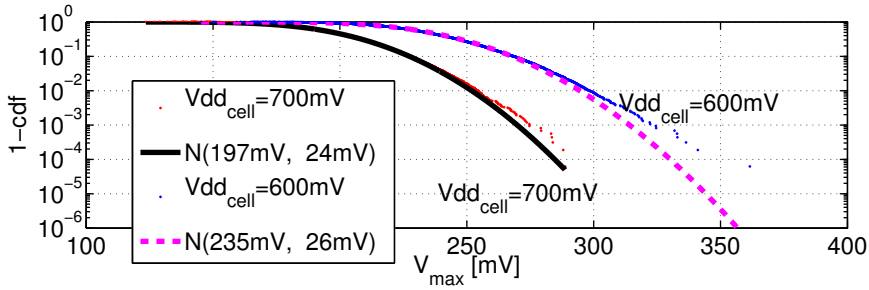
The previous discussion already showed that C_{LBL} has a large impact on the dynamic stability of the cell, as expressed by SNM_{tran}. This section will discuss two more important factors: the impact of the rise time of the word line signal and of the bit line precharge voltage.

Figure 3.30 shows that SNM_{tran} can be improved significantly by increasing the rise time of the word line signal. The rise time for $Wup = 125nm$ is 3 times longer than that for $Wup = 500nm$. This slower rise time improves SNM_{tran} with more than $35mV$ in the region of C_{LBL} between $5fF$ and $30fF$. On the other hand, to obtain the same SNM_{tran}, C_{LBL} can be 2 to 2.5 times larger with the 3 times slower rise time. Figure 3.31 provides a different view on the same data. It might be possible to improve SNM_{tran} for a given delay by optimising the waveform shape of the word line signal, but this will not be analysed here.

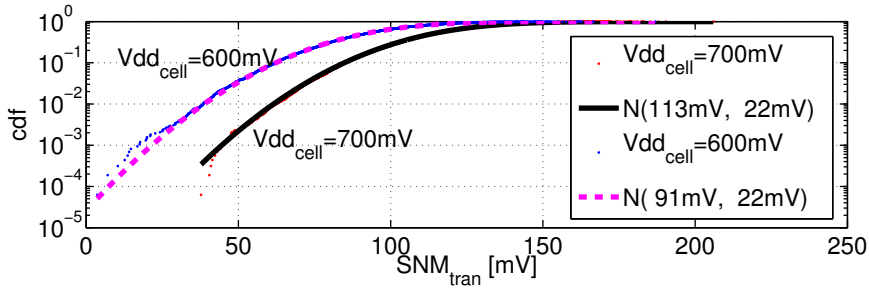
Figure 3.32 shows the impact of the bit line precharge voltage $V_{BL,precharge}$



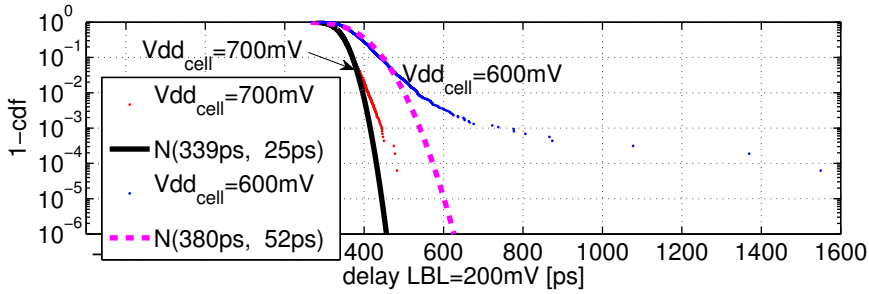
(a) ΔV_{diff}



(b) V_{max}

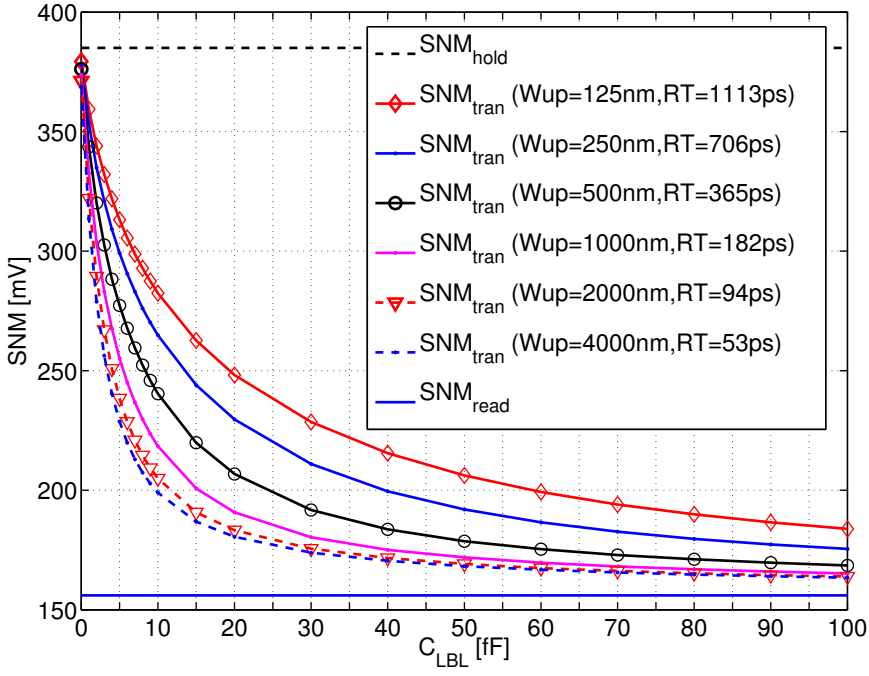


(c) SNM_{tran}

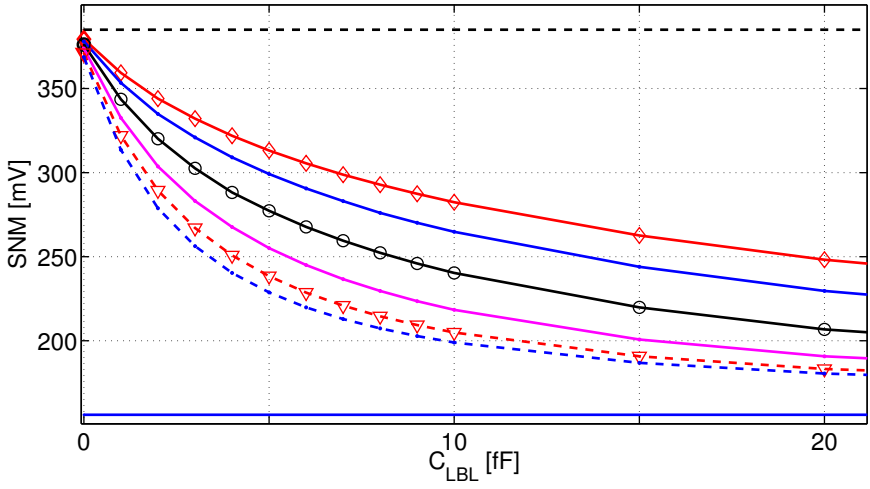


(d) Delay until $V(LBL) = 200mV$

Figure 3.29: Empirical CDFs based on 10000 Monte Carlo runs for $Vdd_{cell} = 700mV$ and $Vdd_{cell} = 600mV$. ($Vdd_{WL} = 1V$, $V_{BL,precharge} = VddBulk = Vdd_{cell}$, $Wup = 1000nm$, $C_{WL} = 24fF$)



(a)



(b) detail of 3.30a

Figure 3.30: SNM_{tran} as function of C_{LBL} and Wup, the width of the SPHVT PMOS transistor in the word line driver. The corresponding word line rise time from 10% to 90% Vdd_{WL} is indicated in the legend. ($Vdd_{WL} = 1V$, $Vdd_{cell} = 1V$, $V_{BL,precharge} = 1V$, $Wdown = 400nm$, $C_{WL} = 24fF$)

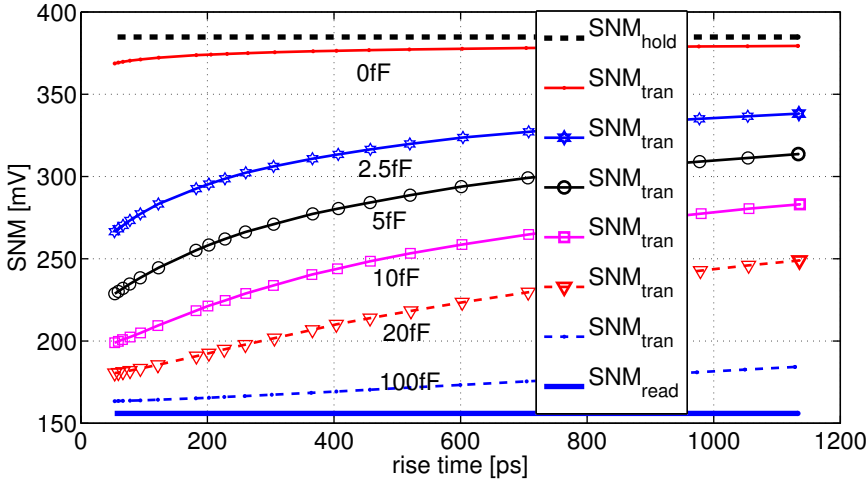


Figure 3.31: SNM_{tran} as function of the word line rise time (10% to 90%) for different values of C_{LBL} as indicated. ($Vdd_{WL} = 1V$, $Vdd_{cell} = 1V$, $V_{BL,precharge} = 1V$)

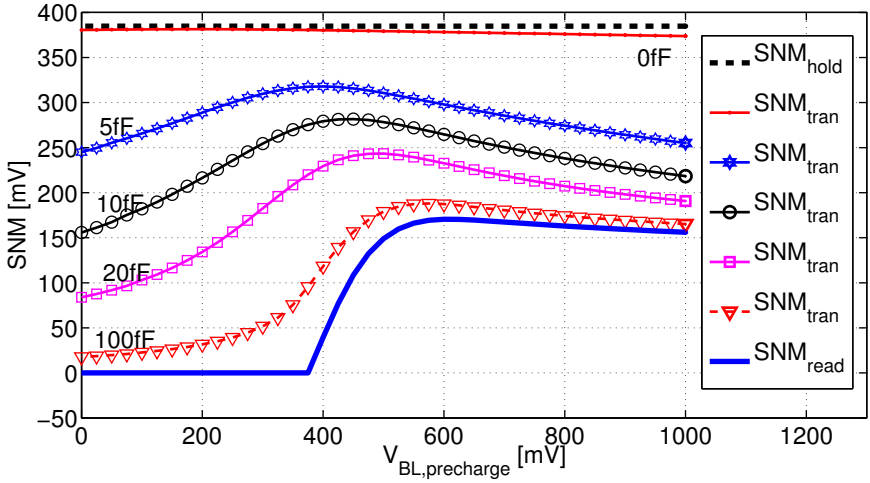


Figure 3.32: SNM_{tran} as function of $V_{BL,precharge}$ and C_{LBL}. ($Vdd_{WL} = 1V$, $Vdd_{cell} = 1V$, $Wup = 1000nm$, $Wdown = 400nm$, $C_{WL} = 24fF$, $RT = 182ps$)

for different values of C_{LBL}. For a bit line capacitance below $10fF$, the optimal precharge voltage for read stability is around $400mV$. Unlike SNM_{read}, SNM_{tran} does not degrade steeply for precharge voltages below the optimum value. Hence, it is safe to precharge short buffered bit lines to $Vdd/2$.

3.7 Summary

This chapter outlines how cells are connected together to build up a local memory matrix and treats the immediate periphery that is needed to access such a local matrix. Techniques that ease the cell requirements receive special attention.

Section 3.2 describes the basic organisation of a local block, with special attention to wire routing, contact sharing and well orientation. This is important information as it limits the type of bias schemes that can be introduced efficiently.

Section 3.3 discusses the local peripheral control circuits. In the traditional memory architecture, this is the local word line driver, but in many state-of-the-art designs that are targeted at technologies with large intra-die variations or at ultra-low voltage operation, additional control wires are introduced, as described in section 2.7. Memories with subdivided bit lines use similar drivers to control the local bit line periphery.

The area, delay and the static and dynamic energy consumption of an example word line driver are presented. Several more interesting implementations are outlined as well.

Section 3.4 introduces an interesting and challenging set of new design techniques that is quickly becoming mainstream as technology moves beyond $65nm$. To aid the cell operation, different bias voltages are applied for the different operation modes. The underlying effects have already been analysed in section 2.5. Section 3.4 also provides an overview of bias schemes that have been discussed in the literature and addresses some of the basic considerations regarding the different bias schemes.

Section 3.5 describes different implementations for the bit line hierarchy, with the local bit line as focal point. This section shows that a good bit line hierarchy can significantly reduce the impact of the cell current I_{read} and the cell's bit line capacitance $C_{BL,cell}$ on the access time and energy consumption of the memory.

Section 3.6 introduces the concept of dynamic read stability, one of the major contributions of this work. With the introduction of short buffered local bit lines, the traditional metric for cell stability during the read operation, SNM_{read}, becomes an excessively pessimistic estimate. This section first shows that cells with large negative SNM_{read} can still be read without causing a read upset. An overview of the recent literature on this topic is provided, which shows that dynamic read stability is now receiving a lot of attention. A novel metric, SNM_{tran}, is introduced to quantify a cell's dynamic read stability as a value

which can be compared directly with SNM_{read} and SNM_{hold}. The section concludes with an analysis of the design options that can be used to improve the cell's dynamic read stability: the number of cells per bit line, the word line rise time and the bit line precharge voltage.

Contributions and highlights

This work introduces the use of dynamic read stability as a new method to relax the cell requirements. A better metric for this dynamic read stability, SNM_{tran}, is also presented.

This text provides the first coherent overview of two other methods that can be used to relax the cell requirements. The use of different bias voltages for the different operation modes can significantly increase the available cell margins, while the impact of the cell read current on the memory delay can be reduced with a better bit line structure and bit line periphery.

Chapter 4

Sense Amplifiers

4.1 Introduction

A *sense amplifier* (*SA*) is a circuit that amplifies a small input signal to a full level output. In more general literature, such circuits are often called comparators. In SRAMs, the most important SAs are the *read sense amplifiers*, which amplify the small voltage swing on the data lines towards the memory output. The main target of the read SAs is to allow the use of a very low swing signal on these data lines. As will be discussed in chapter 6, a smaller required swing has two advantages: it reduces the energy consumed in charging and discharging these large capacitances and it reduces the time the cell or readbuffer needs to develop this swing.

This chapter zooms in on the requirements, the analysis and the design of these read sense amplifiers. The application of sense amplifiers in memories is not limited to this specific function though. They can be used to transmit local address bits in a distributed decoder, they can serve as write receivers or they can improve read performance on the local bit line. In all these cases, the number of instances of these sense amplifiers in a memory is significantly higher than for the read sense amplifiers, hence they are subject to much tighter area constraints. Still, the analyses provided in this chapter can easily be applied to these cases as well.

There are many ways to implement a sense amplifier. They can be categorised based on several properties: strobed or non-strobed, voltage-mode or current-mode and traditional, compensated or calibrated.

Strobed sense amplifiers require a *critical* timing signal, which defines the moment at which time the input signal is sampled and amplified. *Non-strobed amplifiers* do not use such a critical timing signal. Most of them however still use control signals with non-critical timing to switch between precharge and evaluation phases. During the evaluation phase, non-strobed amplifiers have to amplify the input signal continuously (but not necessarily in a linear way).

Voltage mode sense amplifiers consider both the input and the output signal as voltage signals. Sense amplifiers that consider the input signal as a current are referred to as *current SAs*.

Mismatch between transistors puts a lower bound on the required input signal. In *traditional* sense amplifiers, this mismatch is reduced by increasing the size of

the critical transistors. In sense amplifiers with *compensation*, analog feedback is used to cancel out transistor mismatch at run-time, typically using several different phases during the course of a memory access. In *calibrated* SAs, the optimal compensation settings are obtained during a calibration phase. This calibration phase can occur once after fabrication, but it could also be performed every second or each time the temperature has changed by 5 degrees. During normal memory operation, these optimal compensation settings are applied to the SA. Calibration does not require static bias currents.

Most of the recently published SRAMs use strobed voltage-mode SAs. This makes sense, as strobed voltage-mode SAs have quite some advantages. They do not draw static current. They can easily employ positive feedback to provide fast regeneration, and compared to other SAs, their design is rather straightforward.

This chapter discusses traditional strobed voltage mode sense amplifiers. Chapter 5 will discuss SA calibration. Non-strobed SAs, current-mode SAs and offset compensation are outside the scope of this text.

This chapter is organised as follows. First, the important concept of *SA offset voltage* is introduced in section 4.2. This section also discusses the most important "noise" sources that negatively affect the SA operation, including transistor mismatch and thermal noise. Then, the *ecosystem* in which the SA operates is described in section 4.3. Section 4.4 discusses the *important specifications* for SAs. With these specifications in mind, section 4.5 presents a practical simulation-based analysis method for strobed voltage mode SAs. Several implementations of strobed voltage mode SAs are evaluated in section 4.6. Finally, a summary of the findings in this chapter is provided in section 4.7.

4.2 Sense amplifier offset voltage and "noise" sources

In a normal differential amplifier, the offset voltage is the differential input voltage that results in a differential output voltage of zero (when averaged over time). This input voltage is not equal to zero due to transistor mismatch. Every fabricated amplifier instance has its own offset voltage. This offset voltage can be caused in part by systematic effects such as layout asymmetry, but typically, the main cause are uncorrelated intra-die variations. When comparing amplifier designs, their offset distribution can be an important quality metric.

A similar definition of *offset voltage* can be used for strobed SAs.

4.2.1 Offset voltage of strobed SAs

The behaviour of strobed SAs can only be observed when a sensing operation is performed. In this text, we define the offset voltage of a strobed amplifier as follows: the offset voltage V_{offset} is the voltage for which $V_{offset} + \delta$ is sensed as high on average and $V_{offset} - \delta$ is sensed as low on average, where δ is a (small) positive voltage. Figure 4.1 provides a visual representation of this definition. Notice that the offset voltage is distributed symmetrically between the

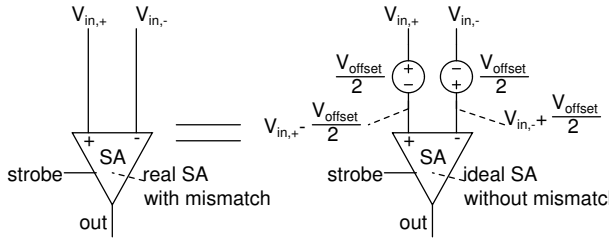


Figure 4.1: Sense amplifier offset voltage V_{offset}.

two input terminals. This has some impact due to the non-linear behaviour of a SA. Due to noise, sensing is not an entirely deterministic process, so the definition includes the phrase "on average". The definition assumes that the chance that the metastable sensing state is not resolved during the sensing period is negligible. This assumes a long enough sensing time, but this is generally only a theoretical issue as actual operation close to the metastable point is avoided in practice.

4.2.2 Parameters impacting SA offset after fabrication

The offset voltage is not entirely fixed after fabrication. It changes with temperature and with bias and supply voltages, and it can change with aging. The impact of these parameters can in theory be observed and quantified by successive measurements, and the value of the SA offset changes accordingly.

In ideal differential SAs, changes in power supply and changes in temperature result in common mode variations which should not affect V_{offset}. However, due to mismatch, there will be some impact. In asymmetric SAs, the impact of power supply and temperature variations should certainly be analysed.

Many sense amplifiers are followed by a static latch, which stores the previously sensed value. If the state of this latch affects V_{offset} of the SA, a history effect is introduced.

Transistor aging can change the SA offset over time. This change often depends on the SA history, especially the ratio of zeros and ones sensed. One interesting cause is *negative bias temperature instability* (NBTI)[Mar09], which causes a shift in PMOS threshold voltage over time. This shift depends on the transistor bias voltages. In [Tak05], a SA is discussed in which this effect results in an additional shift in V_{offset} of $50mV$.

Most of these offset changes are deterministically defined by the underlying phenomena. However, the SA designer often has no control over these parameters. Supply voltages fluctuate and temperature changes. The designer must include sufficient margins against these uncontrolled variations. This requires a separate treatment of each of these uncertainty factors. This is not only needed to avoid excessive design margins, but also to be able to guarantee reliable operation. One especially interesting example is the noise on reference voltages. Systems

node capacitance [fF]	RMS noise voltage [mV]
0.5	2.88
1	2.03
2	1.44
5	0.91
10	0.64
100	0.20

Table 4.1: Representative values for RMS thermal noise voltage at $T = 300K$.

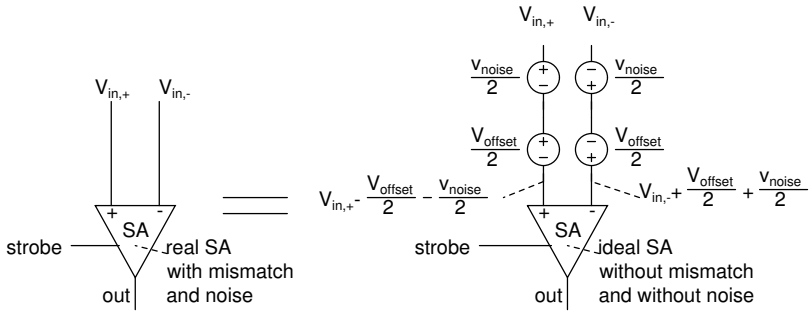


Figure 4.2: Model for strobed sense amplifier including offset voltage V_{offset} to model mismatch and discrete-time input referred noise source v_{noise}. The samples from v_{noise} follow a Gaussian distribution with zero mean and $\sigma = \sigma_{v_{noise}}$.

that use small-signal single-ended signalling typically use a differential SA with one input connected to a reference voltage V_{ref}. In such a setup, all voltage fluctuations which are not common to both the data line precharge voltage V_{prech} and the reference voltage V_{ref} have a 1-on-1 impact on V_{offset}. These effects will not be discussed further in this text, but they certainly need to be kept in mind.

4.2.3 Impact of thermal noise

The sense operation is also influenced by nondeterministic noise sources, such as thermal noise. Because of the "on average" in the definition, this is not reflected in the offset voltage. Still, this noise can result in intermittent SA failure, and appropriate margins must be taken into account.

The root mean square (RMS) thermal noise voltage v_n on a node in a circuit is given by $v_n = \sqrt{\frac{k \cdot T}{C}}$ [San06, Sar93]. Table 4.1 provides some example values. The instantaneous thermal noise voltage amplitudes are described by a Gaussian distribution with $\sigma = v_n$.

It is generally assumed that the impact of thermal noise on the strobed SA behaviour can be modelled as a discrete time system with a Gaussian noise

voltage source v_{noise} at the input. This model is shown in figure 4.2. For every sensing operation, the immediate SA offset is modified by a noise voltage taken from a Gaussian distribution with standard deviation $\sigma_{v_{noise}}$ and zero mean. This assumption is confirmed by measurements[Opr97, Nuz08, Sch07]. Under this assumption, the noise behaviour of a strobed SA is entirely described by $\sigma_{v_{noise}}$. This value can be obtained in one of several ways. It can be measured, it can be obtained from numerical simulations or it can be approximated with analytical calculations. Appendix A provides an in-depth overview of these different approaches.

There is one question that still deserves some attention: is thermal noise actually a relevant issue in strobed SAs as used in memories?

In more traditional analog systems, the performance limitations imposed by mismatch are significantly more strict than those arising from thermal noise [Ste97]. This is true for strobed SAs as well. For example, for a well designed SA with 8x input transistors in 90nm CMOS, typical values for $\sigma_{v_{noise}}$ are between $1mV$ and $3mV$, while $\sigma_{V_{offset}}$ is about $15mV$. However, if thermal noise is not considered during the design, it is possible to end up with a SA with a much worse noise performance. Therefore, it is in general useful to include a simulation of thermal noise somewhere in the design flow as a sanity check.

Thermal noise is also fundamentally more problematic in typical memory SA designs than it is in typical analog blocks, such as analog to digital converters (ADCs). The performance of analog blocks is typically expressed by the RMS value, so sporadic failures are not necessarily a problem. For SAs in typical memories, the situation is quite different: any failure is likely to be catastrophic, as it will often result in corruption of data or instructions.

Reliable memory operation requires that a significant safety margin is dedicated to thermal noise, which translates in a larger required input swing. The question then is: how much margin is needed? This obviously depends on the desired reliability. A simple example will be given without going into too much detail.

A simple (yet rather strict) reliability requirement could be formulated as follows: the SA with the 'worst' mismatch value must have an average failure rate below 1 failure per year. The 'worst' mismatch value could for example be the offset value corresponding to the $1 - 10^{-9}$ CDF value, but the exact choice for this number does not impact the current calculation. If the memory operates non stop at 1GHz, the SA performs $31.5 \cdot 10^{15}$ operations per year. This is the desired mean number of operations before failure, $MNOBF$. The desired failure rate is $fr = \frac{1}{MNOBF}$. The required input voltage margin for thermal noise can then be calculated by using the inverse CDF of v_{noise} and is $V_{required\ input\ margin} = |norminv(fr, \mu = 0, \sigma = \sigma_{v_{noise}})| = 8.4 \cdot \sigma_{v_{noise}}$.

For systems that use traditional SAs, the above reliability requirement is rather on the safe side. However, in systems that use calibration, many SAs will operate close to the edge of the margin that was provided for offset. In such cases, slightly tighter requirements might be needed. For example, if there are 10000

equally critical SAs in the system, and the desired failure rate remains the same, $V_{required\ input\ margin} = |norminv(\frac{fr}{10000}, \mu = 0, \sigma = \sigma_{v_{noise}})| = 9.4 \cdot \sigma_{v_{noise}}$.

When offset calibration or offset compensation are used, the effective impact of SA offset due to mismatch reduces significantly. An improvement with a factor of 5 can be obtained. In this case, thermal noise can easily become the dominant limitation on the required input signal.

4.2.4 Dynamic offset voltage

There is still a problem in the current definition of V_{offset}. V_{offset} is measured with static inputs, hence it is the DC signal that must be applied to overcome mismatch effects. However, we want to use this value to decide how large we should make ΔV_{min}, which is the magnitude of a dynamic signal swing. A ΔV_{min} which is slightly larger than V_{offset} can still result in SA failure, which is not what we had in mind when introducing V_{offset}.

A reasonable solution is to use $V_{offset,dynamic}$ instead, which is determined by using the same rise time as the actual input signal. Now, when ΔV_{min} is slightly larger than $V_{offset,dynamic}$, the SA will operate correctly, as would be expected.

4.3 Sense amplifier ecosystem

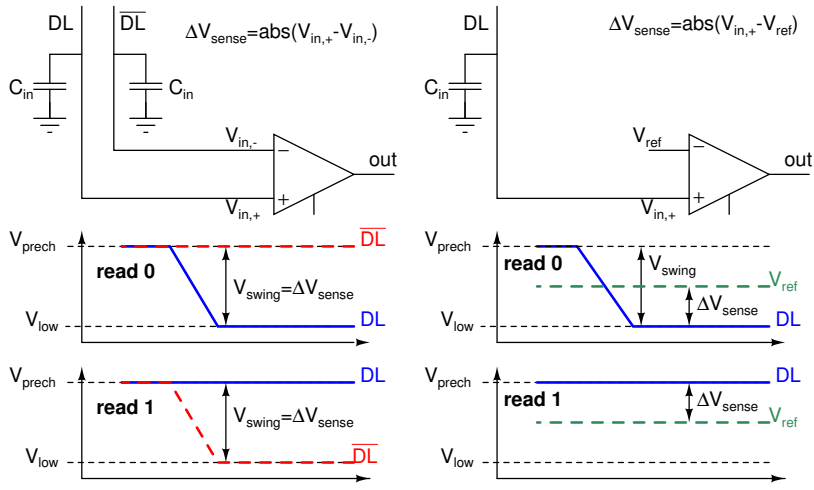
The environment in which the sense amplifier has to operate, has a profound impact on its design. This section discusses the impact of the properties of the input signal and of the output load.

4.3.1 Input signal

4.3.1.1 Signalling method

In SRAMs, the input signal is typically either single-ended or complementary as shown in figure 4.3. More information on the advantages and disadvantages of each signalling type will be provided in 6.2.1. In *complementary signalling*, two data lines are used per bit. They are both precharged to a voltage V_{prech}. During evaluation, one of both lines is discharged to V_{low}. This is a complementary swing $V_{swing,complementary} = V_{prech} - V_{low}$. In *single-ended signalling*, only one data line is used. It is precharged to a voltage V_{prech}. During evaluation, the line is either kept in a high-impedance state or discharged to V_{low}. In the single-ended case, typical sense amplifiers will require a reference voltage V_{ref} as a second input, which should ideally be exactly in the middle between the two possible data-line voltages, V_{prech} and V_{low}. This is a single-ended swing $V_{swing,single-ended} = V_{prech} - V_{low}$.

If the smallest difference between $V_{in,+}$ and $V_{in,-}$ that a given SA design can sense reliably is ΔV_{min}, a complementary signalling system can operate with $V_{swing,complementary} \geq \Delta V_{min}$, while a single-ended signalling system will require twice this swing: $V_{swing,single-ended} \geq 2 \cdot \Delta V_{min}$.



(a) Complementary signalling (b) Single-ended signalling

Figure 4.3: Signalling methods.

4.3.1.2 Other input signal properties

Although the signalling method that is used and the magnitude of the input signal are undoubtedly the most important properties of the input signal for the design of a voltage-mode SA. Some other parameters should be taken into account as well.

The *common mode input voltage* ($V_{CM} = \frac{V_{in,+}+V_{in,-}}{2}$) has a large impact on the operation condition of the input transistors. Values of V_{CM} close to Vdd will require the use of NMOS input transistors, because PMOS transistors with such a high gate voltage would be too slow. Values of V_{CM} close to Vss will likewise require PMOS input transistors.

The input node is a data line, which has a large capacitance to ground. Therefore, the input capacitance added by the SA itself is normally not a significant problem. The capacitive coupling between input nodes and internal nodes of the SA should be considered carefully though, as it can have two detrimental effects which are easily overlooked in standalone SA design.

When a *high-impedance internal node* has a strong, unintended capacitive coupling with an input node, a fast swing at the input will be transferred to this internal node with little attenuation. Typically, this effect will worsen SA operation, which results in a need to use a larger input swing than the voltage difference that is required with static inputs. To ensure the detection of this problem, one should always analyse the SA with representative input voltage waveforms.

The second potential problem is known as *kickback*. The internal nodes of

the SA typically make a full-swing transition. With a coupling capacitance of $2fF$,a total data line capacitance of $200fF$ and an internal swing of $1V$, the capacitive coupling will change the voltage at the input by $10mV$. To ensure detection of potential problems, it would be best to analyse the SA with a representative transmitter connected to the data lines rather than with an independent voltage source. In case of complementary signalling, most of the differential voltage change occurs after the critical sensing is completed, so the impact should be small. Kickback is much more dangerous for single-ended sensing because the two input nodes, the data line and the reference voltage, have a different capacitive load. In this case, the common mode voltage changes on the internal nodes have different impact on the two input nodes.

In this chapter, voltage-mode SAs are studied in general, without considering the exact source of the input signals. Idealised input signals as shown in figure 4.3 are used and are applied to the inputs with ideal independent voltage sources. A rather short transition time of $200ps$ is assumed, so problems with capacitive coupling are detected. Kickback is not taken into account in the numerical analyses of this chapter. Once a specific memory design is selected, the analysis script can easily be adjusted to take the exact source of the input signals into account.

4.3.2 Output load

The output load of the SA has a large impact on its delay. When delay is of the essence, SA sizing can become dominated by delay rather than by mismatch considerations. In this chapter, the scope is limited to the read SAs, which we only load with a small set-reset or clocked latch. Because the capacitances on the sensitive internal nodes of the SA will depend on the value that is stored in the latch, such a state-retaining latch as output load can introduce a history effect in the SA offset.

4.4 Important specifications

Important quality metrics of a SA design are the *required input signal*, the *average dynamic and static energy consumption*, the *'worst-case' delay* and the *area*.

These quality metrics involve averages and worst-case values, so they do not apply to an individual SA instance, but rather to the set of all fabricated SA instances of the SA design. The interpretation of 'worst-case' depends on the desired yield. These quality metrics will be discussed in more detail in this section.

4.4.1 Required input signal

The minimal voltage difference between the two inputs of the sense amplifier that is needed to guarantee *reliable* sensing is called ΔV_{min} in this text. In

this definition, 'reliable' should still be defined. As it is impossible to guarantee 100% correct operation, a yield target will have to be specified. Obviously, ΔV_{min} will be different for different yield targets.

As discussed in 4.3.1.1, ΔV_{min} defines a lower bound for the swing on the data lines, V_{swing}. Reducing this required swing is of the utmost importance to obtain fast and low energy per access memories, as will be discussed in chapter 6.

ΔV_{min} must be large enough to guard against many different imperfections and uncertainties. There are at least three important contributors to ΔV_{min}: *transistor mismatch* ($\Delta V_{min,mismatch}$) *thermal noise* ($\Delta V_{min,noise}$) and a *required overdrive voltage* to limit SA delay ($\Delta V_{min,overdrive}$). The different contributors to ΔV_{min} will be discussed now.

4.4.1.1 Transistor mismatch

$\Delta V_{min,mismatch}$ is in normal circumstances the single most important contribution to ΔV_{min}. It depends on technology, SA design and sizing, as well as on the desired yield level.

A typical reliability requirement could be to allow one die out of 10000 to fail due to SA malfunctioning. If the system contains 1000 SAs, this reliability requirement requires that at most one SA out of 10M fails. The allowed failure rate fr is $\frac{1}{10\mathtt{M}}$. Failures happen at both sides of the offset distribution, so each side can allow a failure rate of $\frac{fr}{2}$. If V_{offset} follows a Gaussian distribution with zero mean and a standard deviation $\sigma_{V_{offset}}$, the required $\Delta V_{min,mismatch}$ can be calculated as $\Delta V_{min,mismatch} = |norminv(\frac{fr}{2}, \mu = 0, \sigma_{V_{offset}})| = 5.33\sigma_{V_{offset}}$. The desired yield target is often expressed in the number of standard deviations, n. In the above example, $n = 5.33$.

In a well designed SA, the SA offset voltage is dominated by the mismatch in threshold voltage between the two input transistors. When a reliability requirement is chosen, the Pelgrom constant of the input transistors provides a good estimate of the relation between $\Delta V_{min,mismatch}$ and the size of the SA input transistors.

$$\Delta V_{min,mismatch} \geq n \cdot \sigma_{V_{offset}} \approx \frac{n \cdot A_{\Delta V_T}}{\sqrt{W \cdot L}} \tag{4.1}$$

The upscale factor S is defined as $S = \frac{W \cdot L}{W_{min} \cdot Lmin}$, The standard deviation of the difference in threshold voltage ΔV_T between two minimal transistors is $\sigma_{\Delta V_{T,min}} = \frac{A_{\Delta V_T}}{\sqrt{W_{min} \cdot Lmin}}$. With these definitions, the previous statement can be reformulated as

$$\Delta V_{min,mismatch} \geq \frac{n \cdot \sigma_{\Delta V_{T,min}}}{\sqrt{S}} \tag{4.2}$$

$$S \geq \left(\frac{n \cdot \sigma_{\Delta V_{T,min}}}{\Delta V_{min,mismatch}} \right)^2 \tag{4.3}$$

Remember that $\Delta V_{min,mismatch}$ is the amount of input signal allocated as a safety margin against offset due to transistor mismatch. The dynamic energy

	allowed SA failure rate:		1 in 10^3	1 in 10^4	1 in 10^6	1 in 10^9
S	$\sigma_{\Delta V_T}$ [mV]	required ΔV_{min} :	$3.3\sigma_{\Delta V_T}$	$3.9\sigma_{\Delta V_T}$	$4.9\sigma_{\Delta V_T}$	$6.1\sigma_{\Delta V_T}$
1	38.5	ΔV_{min} [mV]	127	150	189	235
2	27.2	ΔV_{min} [mV]	90	106	133	166
4	19.3	ΔV_{min} [mV]	64	75	94	117
8	13.6	ΔV_{min} [mV]	45	53	67	83
16	9.6	ΔV_{min} [mV]	32	38	47	59
32	6.8	ΔV_{min} [mV]	22	27	33	42
64	4.8	ΔV_{min} [mV]	16	19	24	29

Table 4.2: Minimal $\Delta V_{min,mismatch}$ that is required for different yield targets and upscale factors S. Only mismatch in the threshold voltage of the input transistors is considered. [$A_{\Delta V_T} = 4mV \cdot \mu m$, $\sigma_{\Delta V_{T,min}} = 38.5mV$]

consumption E_{SA} of a typical strobed SA is almost directly proportional to S, so for a fixed input swing, $E_{SA} \propto n^2$. This formula gives a good first estimate of the cost associated with setting too rigid reliability margins: designing for $n = 6.1$ ($fr = 10^{-9}$) rather than for $n = 4.9$ ($fr = 10^{-6}$) increases SA energy to $\frac{6.1}{4.9}^2 = 155\%$. Obviously, one should also consider the option to increase the input swing.

Table 4.2 provides some numerical examples obtained by these equations for a commercial 90nm technology with $A_{\Delta V_T} = 4mV \cdot \mu m$, $W_{min} = 120nm$ and $L_{min} = 90nm$. According to Pelgrom's law, $\sigma_{\Delta V_{T,min}} = 38.5mV$.

It is interesting to compare $A_{\Delta V_T}$ for the available transistors in the technology. Typically, PMOS threshold matching is slightly better in current technology nodes, e.g. $3.5mV\mu m$ versus $4mV\mu m$. This might or might not weight up against the NBTI which affects PMOS transistors.

4.4.1.2 Thermal noise

$\Delta V_{min,thermalnoise}$ guards against fluctuations in effective SA toggle point due to thermal noise. This topic was treated in some detail in 4.2.3.

This margin is often forgotten in SA design for memories, but has received some attention in the design of comparators for other applications [Nuz08]. Typical reported values for $\sigma_{v_{noise}}$ for a SA with 8x input transistors in 90nm range from $1mV$ to $3mV$. A reasonable safety margin is $\Delta V_{min,thermalnoise} = 8.4 \cdot \sigma_{v_{noise}}$. This margin is often small compared to that required by transistor mismatch. However, when calibration or compensation are used to reduce the impact of transistor mismatch, the margin for thermal noise can become dominant.

4.4.1.3 Required overdrive voltage

The required $\Delta V_{min,overdrive}$ depends on the desired worst-case speed. The impact of the effective input signal on the delay of strobed SAs will be discussed

in more detail in section B.

For regenerative SAs, the delay depends logarithmically on the required gain $A = \frac{v_{out,required}}{v_{in,effective}}$: $delay = t_0 + \tau \ln A = t_0 + \tau_{10} \log_{10} A$. The parameters t_0 and τ depend on the SA implementation and $\tau_{10} = \tau \cdot ln(10)$. The worst-case value for $v_{in,effective}$ is $\Delta V_{min,overdrive}$. Thanks to the logarithmic dependence, a small $\Delta V_{min,overdrive}$ is often acceptable. In [Sch07], τ_{10} was $44ps$ per decade.

In absence of positive feedback, delay is much more sensitive to the effective input voltage, directly proportional in the best case: $delay = t'_0 + S_0 \cdot A$. In this case, a large required amplification A has a huge impact on delay and should be avoided.

4.4.2 Dynamic energy consumption

For low energy per access memories, the dynamic energy consumption is probably the second most important metric. Especially for memories with a wide word length, read SA energy consumption does matter.

The dynamic energy consumption of the SA is the energy associated with its active operation. It is the increase in SA energy consumption compared to an idle cycle. When different SA designs are compared, it is important that the energy calculation is accurate to ensure a fair comparison. Special care is needed to include the energy that is needed to generate the SA control signals. The dynamic energy consumption should also include the input capacitance of the sense amplifier, although this can often be neglected because of the low swing on the data lines. The charge that flows to the input gates of the next stage is typically included implicitly in the SA energy consumption. As long as identical loads are used for all studied SAs, this has no impact on conclusions.

4.4.3 Static energy consumption

Static energy consumption can consist of bias currents or leakage currents. Almost all SAs studied avoid the use of constant bias currents. Notable exceptions are current SAs and amplifiers with offset compensation. For memories operating at reasonable speeds, leakage energy due to the global read SAs should be small compared to their dynamic energy consumption.

4.4.4 Delay

A good definition for the delay of a SA is more complicated than it might look at first.

For non-strobed SAs, a similar definition can be used as for normal gates: the SA delay is the time it takes between the arrival of the input signal and the moment the SA output is ready. All that is left is to define “arrival” and “ready”, and different SA designs (operating in the same ecosystem) can be compared.

For strobed SAs, the situation is fundamentally different, as there are three important events now. There is the arrival time of the input signal (t_{in}), the

strobe time (t_{strobe}) and the moment at which the SA output is available (t_{out}). This situation is very similar to that of a normal flipflop.

A delay definition based on t_{in} and t_{out} only is obviously flawed as this delay changes with t_{strobe}. A definition based on the delay from strobe signal to output is tempting, but rather useless. If this delay would bother us, we could simply generate the strobe signal a bit earlier.

A more interesting delay when comparing different strobed SAs, is the smallest $t_{out} - t_{in}$ that can be obtained with each SA. For this, the strobe time t_{strobe} needs to be optimised for each SA design individually. This is rather easy to implement under nominal conditions, without uncorrelated variations. The same input signal is used for each SA design under test. For each SA design, the strobe timing that results in the smallest $t_{out} - t_{in}$ is obtained using spice simulations. The delay of the strobed SA is then $delay = t_{out} - t_{in}$ at the optimal t_{strobe}. In setups where the SA is strobed by the system clock, $t_{setup} = t_{strobe} - t_{in}$ or $t_{develop} = t_{out} - t_{strobe}$ might be more important than the total SA delay.

Things get more complicated when uncorrelated intra-die variations are considered. t_{in} will be different for different memory cells, and different instantiations of the same SA design will have different setup times due to their different offset voltages V_{offset}. Still, all SAs in the same memory will be strobed at the same time. In this text, a very simple approach is used. "Worst-case values" are assumed for both the input waveform and t_{setup}. These worst-case values are defined by their respective distributions and by the desired reliability, not as in traditional corner-based design. This approach is certainly safe, but it does result in over design. A good hands-on approach to manage this analysis in a more accurate way has been discussed in [Hou08].

With uncorrelated variations, $t_{develop}$ and t_{out} become stochastic variables with corresponding distibutions. It depends on the memory design which property of this distribution matters most. If the delay variations can be averaged out over de next few logic stages, the mean delay is crucial, and the standard deviation only makes things a little bit worse. If the next stage is a flipflop, the situation is different. There is no averaging out of the delay, and the worst-case value of t_{out} is what matters most.

4.4.5 Area

Because our designs require only one set of read SAs, their area is not that important.

In other settings, the area consumption is often considered to be critical. This is true for read SAs in designs that require one SA per bit column. It is certainly true for SAs that function as write receivers, as more than one write receiver is needed per bit column.

4.5 Analysis of strobed voltage-mode SAs

This work relies mainly on spice simulations to assess the quality of SAs. The main advantages of this approach are that all desired and undesired effects are taken into account and that it is relatively easy to test different SA designs. The main disadvantages compared to analytical calculations are the risk of a reduced amount of design insight and the longer simulation time.

Simulation time is no longer necessarily a problem. On a recent desktop computer, it takes about 0.2 seconds to simulate a SA sensing operation and extract the energy and delay from strobe to output. To approximate the SA offset of a sense amplifier instance to an interval of $V_{interval} = \pm 1mV$ using a simple binary search requires $2 + \log_2 \frac{V_{range}}{V_{interval}} \approx 14$ steps. This results in a total analysis time per instance of about 3 seconds. Searching the optimal strobe time is of a similar time complexity.

The loss of design insight could be a more fundamental problem. To counter this, appendix B provides an overview of the literature on the analytical analysis of SAs, as well as a first-order symbolic analysis for the most important SAs. The best solution is obtained by using the insights obtained from these symbolic analyses to understand and improve the accurate results obtained with simulations.

This section discusses the analysis of SA specifications based on simulations. The specifications that will be discussed are the distribution of the offset voltage, the delay, the dynamic energy and the noise behaviour.

4.5.1 Analysis of the offset distribution

The most important step in the analysis of a SA is to calculate the distribution of its offset voltage. Unless specified otherwise, the analyses in this text rely on $V_{offset,dynamic}$.

A workable method to obtain an approximation of this distribution, is to use a sensitivity based approach. The SA contains devices with parameters that suffer from variations. Each transistor suffers from V_T and β mismatch. In theory, capacitances also suffer from variations, but they will not be included here as they seem to be less important. If there are N transistors in the SA, there are $2 \cdot N$ parameters P_i.

If V_{offset} is a linear function of changes ΔP_i in the parameters, $V_{offset} = \sum_{i=1}^{2N} S_i \cdot \Delta P_i$ with S_i the sensitivities that can be calculated as $S_i = \frac{\partial V_{offset}}{\partial P_i}$. If each parameter follows an independent Gaussian distribution with standard deviation σ_i, V_{offset} follows a Gaussian distribution with $\sigma_{V_{offset}} = \sqrt{\sum_{i=1}^{2N} {S_i}^2 \cdot {\sigma_i}^2}$. With $S'_i = S_i \cdot \sigma_i$, this becomes $\sigma_{V_{offset}} = \sqrt{\sum_{i=1}^{2N} {S'_i}^2}$. S'_i is the amount of change in V_{offset} for $\Delta P_i = \sigma_i$.

In reality, these assumptions need to be verified first. A reasonably solid verification can be obtained by sweeping each parameter individually over a range

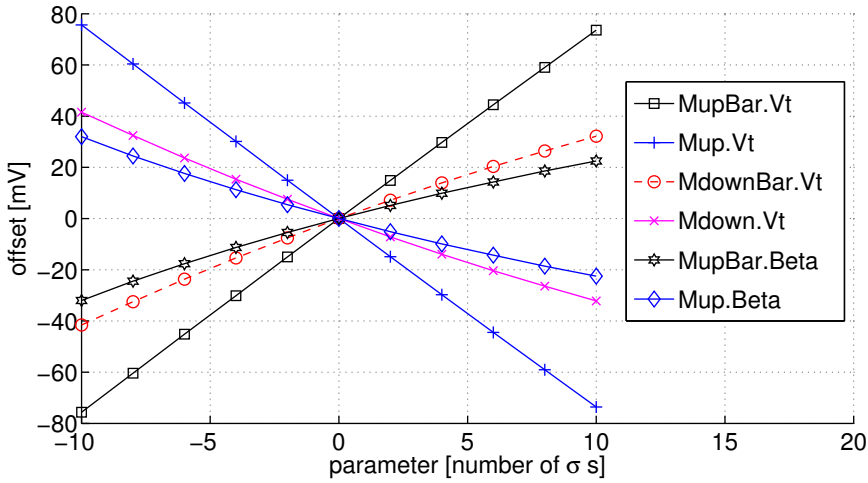


Figure 4.4: V_{offset} as function of the important mismatch parameters. The slope of these curves yields S'.

that is slightly wider than what is actually needed, e.g. from $-8\sigma_i$ to $8\sigma_i$. If V_{offset} behaves linear over this entire range, the above assumptions can be maintained. If there are some slight deviations, one can replace the sensitivity with the worst case linear approximation over the interval. In this case $\sigma_{V_{offset}}$ will be overestimated slightly and the resulting design choices are on the safe side. If the deviations are larger, the SA design will have to be changed, or a more accurate design approach needs to be developed.

Figure 4.4 shows the results of such a sweep for the SA of figure 4.5. All parameters were swept, but only the most important parameters were plotted. In this case, the linear approximation is certainly acceptable for the sweeps of the V_T of Mup and MupBar. The sweeps of the other important parameters are slightly curved. However, they are still rather well behaved, and do not become dominant over the interval of interest, e.g. $[-8\sigma_i, 8\sigma_i]$. To be on the safe side, we replace the sensitivities with the worst linear approximation that can be made in the interval of interest. In casu, this means that we will replace the sensitivity to the V_T of transistor Mdown with $S'_i = \frac{32.5mV}{-8}$.

Once the assumption of linearity has been verified, the results can be summarized as in table 4.3. The total overestimated $\sigma_{V_{offset}}$ is $13.32mV$ Remember that the mismatch parameter associated with beta is $\frac{\beta}{\beta_0}$, which is expressed as a percentage. Notice that it is $\sqrt{\sum_{i=1}^{2N} contribution_i{}^2}$ that must sum to 100%. The square in this formula further reduces the impact of the less important parameters. If $\sigma_{V_{offset}}$ for the example of table 4.3 is estimated based on Mup and MupBar alone, $\sqrt{7.556mV^2 + 6.544mV^2} = 10.7mV$. That is 80.6% of the real value $13.32mV$. In exploratory design phases, it is typically acceptable to only consider the mismatch of the input transistors.

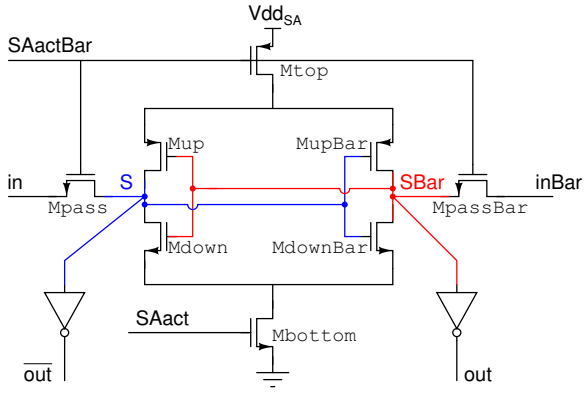


Figure 4.5: The schematic of the SA used in the example. All transistors are SPLVT and have minimal length ($L = 80nm$). Widths are $Wbottom = 800nm$, $Wpass = 400nm$, $Wup = 1080nm$, $Wdown = 1080nm$ and $Wtop = 800nm$.

device	param	S' $[\frac{mV}{\sigma}]$	contri-bution	Width [nm]	Length [nm]	σ_i
MupBar	Vt	7.556	57%	1080	80	8.9mV
Mup	Vt	7.544	57%	1080	80	8.9mV
MdownBar	Vt	4.056	31%	1080	80	16.4mV
Mdown	Vt	4.053	30%	1080	80	16.4mV
MupBar	Beta[a]	3.076	23%	1080	80	2.6%
Mup	Beta	3.064	23%	1080	80	2.6%
Mpass	Beta	1.807	14%	400	80	8.7%
MpassBar	Beta	1.794	13%	400	80	8.7%
Mdown	Beta	1.587	12%	1080	80	5.3%
MdownBar	Beta	1.575	12%	1080	80	5.3%
MpassBar	Vt	0.286	2.0%	400	80	26.9mV
Mpass	Vt	0.256	1.9%	400	80	26.9mV
Mbottom	Beta	0.049	0.4%	800	80	6.1%
Mtop	Vt	0.049	0.4%	800	80	10.3mV
Mbottom	Vt	0.049	0.4%	800	80	19.0mV
Mtop	Beta	0.024	0.2%	800	80	3.1%
	$\sigma_{V_{offset}}$:	$13.32mV$	100%			

Table 4.3: Overview table of a linear sensitivity analysis for V_{offset}.

[a] the beta parameter refers to $\frac{\beta}{\beta_0}$.

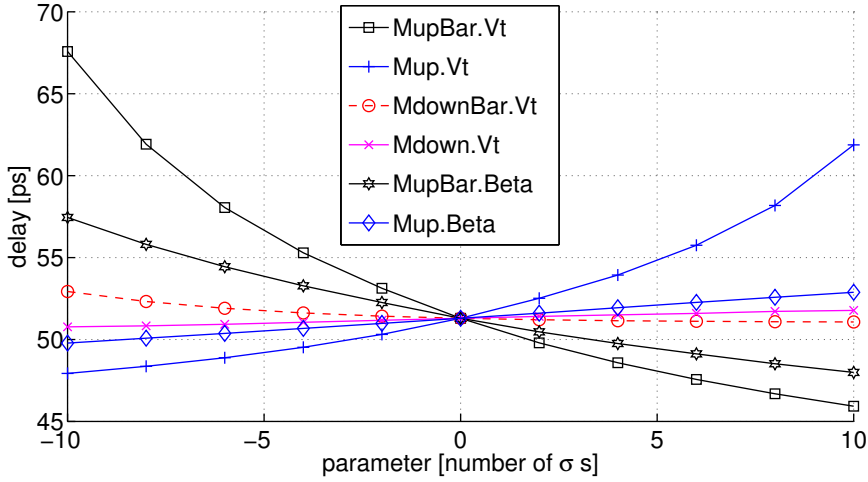


Figure 4.6: SA delay as function of the important mismatch parameters.

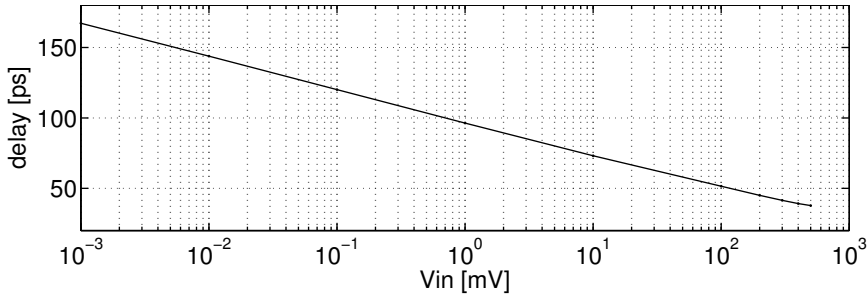


Figure 4.7: SA delay as function of V_{in}.

4.5.2 Analysis of SA delay

The delay used in this analysis is the delay from the time the strobe signal reaches $0.5 \cdot Vdd$ until the output of the output inverters reaches $0.5 \cdot Vdd$. As discussed above, better definitions of SA delay exist. However, as SA delay is not the main problem in this work, this simple measure can suffice for comparisons. Only the delay for the read-0 operation is considered here: signal in is pulled down from $100mV$ to $0mV$ while signal $inBar$ remains at $100mV$. This must be accounted for when performing yield calculations.

The sweep based approach that was used to analyse the offset distribution can also be used for the delay. Figure 4.6 provides the sweep results for the SA from figure 4.5. These sweeps provide quite some useful information and can be used to construct a suited estimation of the delay distribution. In this case, when the worst linear approximations on the interval $[-8\sigma, +8\sigma]$ is used for each parameter P_i, this gives $\sigma_{delay} = 1.9ps$. Due to the strongly non-linear behaviour of delay as function of the mismatch parameters, this number should

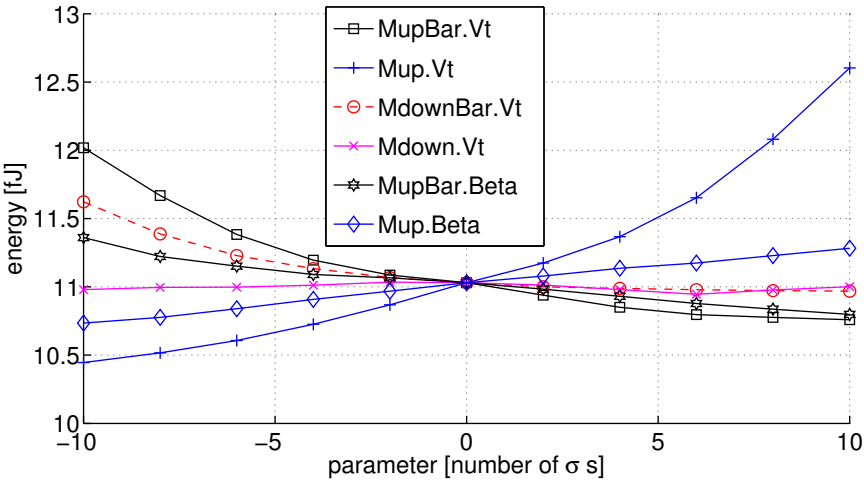


Figure 4.8: SA dynamic energy consumption as function of the important mismatch parameters.

be treated with sufficient care.

During the design process, a question of interest is how much of the input signal should be reserved for $\Delta V_{min,overdrive}$ to obtain a reasonable speed. An easy, approximate answer can be obtained by performing a simulation for different input swings, as in figure 4.7. For input swings below $200mV$, the delay dependency is logarithmic, as the analytical analyses predict. The delay can be written as $delay = t_0 + \tau_{10} log_{10} \frac{100mV}{V_{in}}$, with $t_0 = 51ps$ and $\tau_{10} = \frac{23.1ps}{decade}$. With a delay target of $100ps$, the lowest acceptable $\Delta V_{min,overdrive}$ is around $1mV$. This value is very useful for hand calculations, but because SA behaviour can be strongly non-linear, the final SA design should still be validated with a sweep analysis as in figure 4.6.

4.5.3 Analysis of SA energy

The strobed SAs studied in this chapter do not use static bias currents, so in typical circumstances, the SA energy consumption is dominated by dynamic energy consumption. The dynamic energy is obtained by integrating the currents for one entire read-"0" cycle, including both activation and de-activation. The gate energy for the control transistors is included in this figure. The energy that is consumed by the gate of the output inverters is included, but the energy associated with the I_{ds} currents of the output inverters is excluded.

Figure 4.8 shows the dynamic energy consumption for read-"0" as a function of the different parameters. This graph can be interesting to detect potential design hazards, but the standard deviation of the dynamic energy distribution is much less important than that for offset and delay, as energy consumption averages out over different instances. In this case, the worst linear approximations on

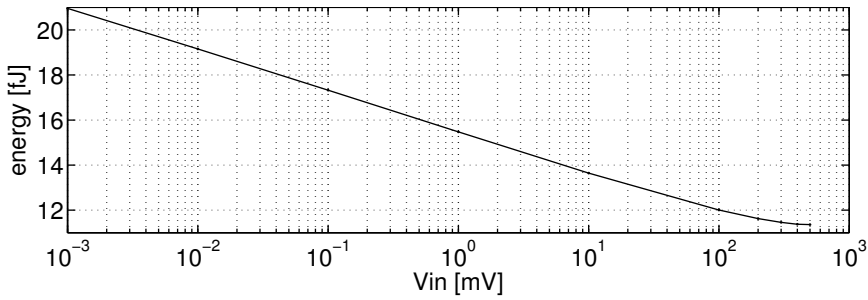


Figure 4.9: SA dynamic energy consumption as function of V_{in}.

the interval $[-8\sigma, +8\sigma]$ gives $\sigma_{energy} = 0.18fJ$.

Figure 4.9 shows the impact of the input swing V_{in} on the dynamic energy consumption. Because short-circuit current is flowing during the entire period that the SA is in a state close to the meta-stable state, the energy follows a similar curve as the delay. For this SA, the slope is $\frac{1.8fJ}{decade}$. This information should be kept in mind when selecting the nominal input swing. It has little impact on the choice of $\Delta V_{min,overdrive}$ as few SA will actually operate at this effective input voltage.

4.5.4 Analysis of SA noise behaviour

The numerical simulation method described in appendix A is used to assess the impact of thermal noise on the different SAs.

4.6 Example SA implementations

In this section, five different SA implementations are analysed. Their basic operation is explained and peculiar effects are highlighted. To enable a fair comparison, each of the SAs is designed for the same specifications. They are all loaded with the same small inverter outputs, with NMOS $W = 200nm$, PMOS $W = 400nm$. A complementary input signal $\Delta V_{min} = 100mV$ is used. Both inputs are precharged at $100mV$, and one of the inputs is pulled down to $0V$ in a timespan of $200ps$. A SA failure rate $fr = 10^{-9}$ ($n = 6.1$) and a delay below $100ps$ are targeted. To limit the complexity, the comparison itself is performed at a fixed supply voltage of $Vdd = 1V$.

4.6.1 Drain-input latch-type SA

4.6.1.1 Operation and analysis at $1V$

Figure 4.10 shows the schematic of the drain-input latch-type SA. This SA has been in use for quite a while [Sak93].

Figure 4.11 shows the nominal waves for the drain-input latch-type SA with setting S1 from table 4.4.

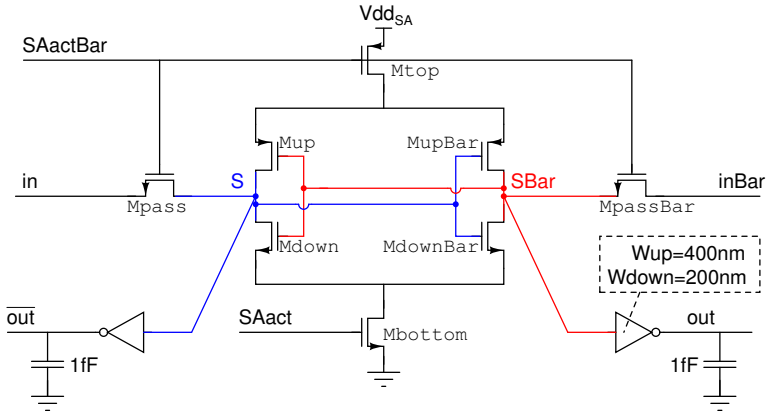


Figure 4.10: The schematic of the drain-input latch-type SA. All transistors have minimal length ($L = 80nm$).

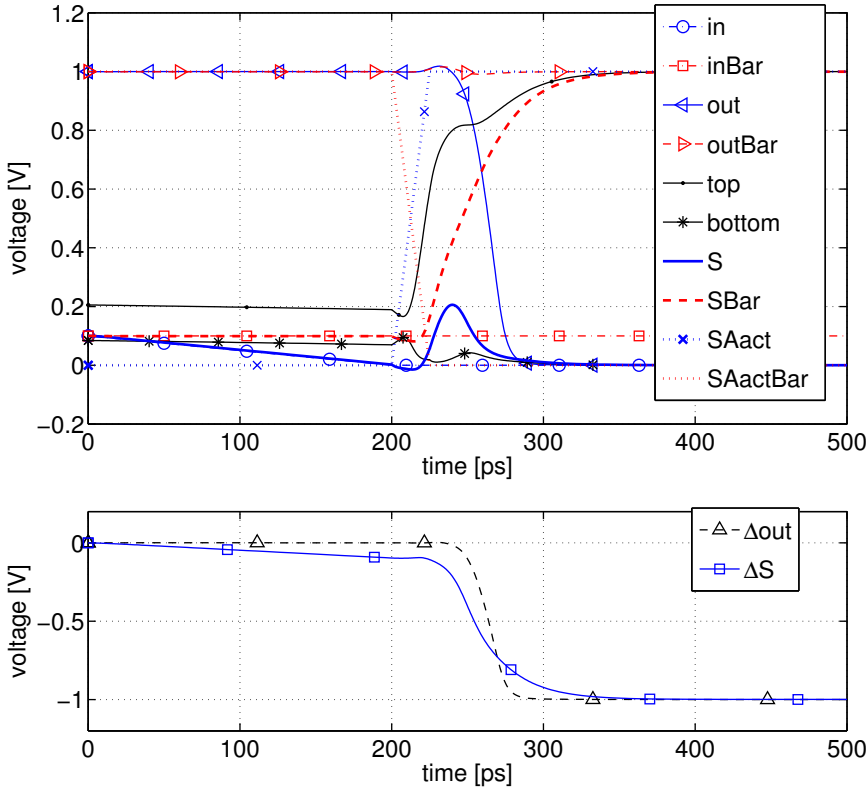


Figure 4.11: The nominal waveforms for the drain-input latch-type SA with settings S1.

setting	S1	S2	S3	S4
Mtop W [nm]	800	800	800	800
Mup W [nm]	1080	1080	1080	1080
Mdown W [nm]	1080	200	200 (HVT)	600
Mbottom W [nm]	800	200	200 (HVT)	200
Mpass W [nm]	400	200	200	200
nominal delay [ps][a]	51.4	41.7	42.2	45.3
σ_{delay} [ps]	1.9	1.8	2.4	1.7
nominal energy [fJ]	11	7.4	7.3	8.3
τ_{10} $[\frac{ps}{decade}]$	23.1	26	> 40	24.2
$6.1 \cdot \sigma_{V_{offset}}$ [mV]	81.1	107.3	106.1	88.5
$\sigma_{V_{offset}}$ [mV]	13.3	17.6	17.4	14.51
contribution to $\sigma_{V_{offset}}$				
by Mup(Bar).Vt [mV]	7.56	8.7	10.1	8.03
by Mdown(Bar).Vt [mV]	4.06	7.4	4.9	4.94

Table 4.4: Overview table of different SA settings that were used for the drain-input latch-type SA. Reported performance is at $Vdd = 1V$. All transistors are SPLVT, unless specified otherwise.

[a] All nominal values are simulated at $V_{in} = 100mV$

The basic operation is as follows. As long as $SAact$ is high and $SAactBar$ is low, the pass transistors Mpass and MpassBar connect the input with the internal nodes S and $SBar$. When input signal in is pulled low, $SBar$ follows. With setting S1 and a $200ps$ input fall time, there is a voltage drop of $2.3mV$ over the pass transistor. This reduces the effective input signal that is available for the actual amplifying latch. Mtop and Mbottom act as switched current sources, which are disabled in this first phase. The second phase starts when $SAact$ becomes high and $SAactBar$ becomes low. The pass transistors disconnect the internal nodes from the inputs. Current source Mtop pulls up the *top* node, which is the source for Mup and MupBar which then start to conduct current. In this phase, the PMOS transistors effectively act as a differential amplifier with positive feedback: node S is connected to the gate of MupBar. S is at a lower voltage than $SBar$, so MupBar will draw more current than Mup, which causes $SBar$ to increase faster than S, quickly amplifying the existing difference. In the third phase, $SBar$ exceeds the threshold voltage of MdownBar, and node S is actively pulled down. When S is at $0V$ and $SBar$ is at Vdd_{SA}, all current paths are disabled. When $SBar$ exceeds the threshold of the output inverter, *out* goes down quickly. When $SAact$ and $SAactBar$ are restored to their original values, the current sources get disabled and the pass transistors connect the internal nodes with the inputs. At this time, quite some charge flows from $SBar$ into the input $inBar$. Nominal delay (strobe 50% to output 50% is $51.4ps$ and energy consumption is $11fJ$.

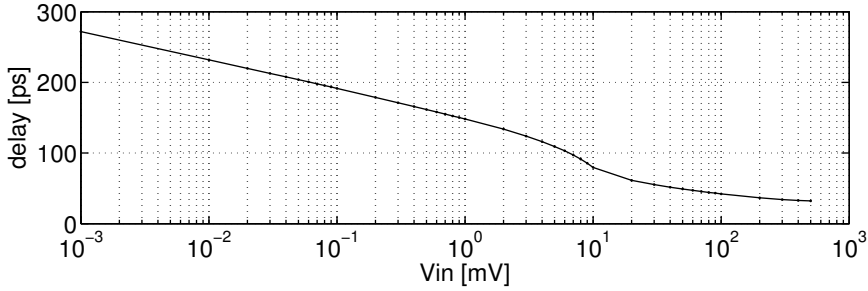


Figure 4.12: Delay as function of V_{in} for latch-type SA, setting S3.

Notice that there are some voltage bumps on the internal nodes S, $SBar$, *top* and *bottom*. A plot of $\Delta S = S - SBar$ shows more clearly what the exact impact on the sensing operation is. Due to the large transients (including the transients on *top* and *bottom*) on the non-linear gate capacitances, part of the input signal gets lost before actual sensing starts. At $100mV$ input swing, $3mV$ is lost, at $10mV$ input swing it is only $0.3mV$, so this is about 3%. This effect was studied in [Sin04].

In this SA, the latest time of data arrival is when the pass transistors disconnect the inputs. However, it takes until the top node is pulled up to about $0.8V$ before the fast amplification starts. This problem could be alleviated by activating Mtop about $20ps$ before disabling the pass transistors, which results in an effective delay reduction (latest time of data arrival to output time) of about $10ps$. This will not be considered further.

Another improvement that could be used is to remove transistor Mbottom. Because the precharge voltage that is used is very low, the static subthreshold currents through Mdown and MdownBar are probably acceptable. It might be advisable to increase their threshold voltages though. This will also not be considered further.

This SA with setting S1 was used in section 4.5.2, so figures 4.4, 4.6, 4.8, 4.7 and table 4.3 apply. The most important specs are summarised in table 4.4.

Sizing S1 is far from optimal. The NMOS transistors contribute significantly to the capacitance on the internal nodes and hence to the energy consumption. Their large size does not improve speed, as the additional capacitive loading is more important than keeping S low. As table 4.3 indicates, their contribution to $\sigma_{V_{offset}}$ is still limited.

Sizing S2 tries a very aggressive setting. All NMOS transistors get $W = 200nm$. The most important specs are again summarized in table 4.4. These results indicate a significant improvement in energy consumption and speed at the cost of mismatch: with this sizing, the bottom NMOS transistors become important in the mismatch budget.

As $V_{offset} \approx \frac{g_{m,n} \cdot v_{osn} + g_{m,p} \cdot v_{osp}}{g_{m,n} + g_{m,p}}$ (see equation B.1), it would seem that reducing

$g_{m,n}$ should reduce the impact of the NMOS mismatch on the offset voltage. Setting S3 uses SPHVT transistors for the bottom NMOS transistors to try to obtain this by reducing Ids. This gives some interesting results: the impact of mismatch in Mdown(Bar) is reduced as expected, but at the same time, the impact of mismatch in V_T of Mup(bar) on $\sigma_{V_{offset,dynamic}}$ went up to a value which is 14% larger than the σ_{V_T} itself. Below an effective input voltage of $10mV$, delay goes up steeply, as shown in figure 4.12. This was obviously not the desired effect.

Setting S4 reduces $6.1\sigma_{V_{offset}}$ to below $100mV$, as desired for our comparison.

The remainder of this section will discuss supply voltage scaling for the SA and the impact of the width of transistor Mtop.

4.6.1.2 Supply voltage scaling

When delay slack is available, it is very interesting to operate the SA at a reduced supply voltage as this greatly reduces dynamic energy. Figure 4.13 shows the results for setting S4, using all SPLVT transistors. Strange effects on the graphs at low supply voltages, such as the offset introduced by Mtop at $0.3V$, are artifacts due to the fact that the post-processing has problems with sensing events that are not finished within the activation time. Down to $450mV$, SA offset hardly degrades, and the SA operates as desired. At $450mV$, the nominal delay is $250ps$ and the estimated $\mu+6.1\sigma$ delay is $500ps$. The energy consumption (calculated as $Q \cdot Vdd$) is reduced with a factor 5.8x, from $8.3fJ$ to $1.44fJ$. This energy reduction is slightly larger than the factor $\left(\frac{1V}{0.45V}\right)^2$ that is expected for normal CMOS gates because of the short-circuit currents involved. Below $450mV$, $\sigma_{V_{offset}}$ goes up slowly, and delay increases quickly. At $350mV$, $\sigma_{V_{offset}}$ is $21mV$ instead of $14.5mV$, and the SA does no longer obtain the desired yield with an input swing of $100mV$. When the failing points are dropped, nominal delay is $800ps$ and $\sigma_{delay} = 280ps$.

Scaling down the supply is significantly tougher when higher threshold transistors have to be used. When the same transistor setting S4 is used, but all transistors are changed from SPLVT to SPHVT, delay already increases significantly at $650mV$, as is obvious from figure 4.14. Especially σ_{delay} is problematic. At this voltage, energy is reduced from $8.1fJ$ to $3.1fJ$, a reduction with only a factor 2.6x.

4.6.1.3 Width of switched current source Mtop

Another interesting question is what the impact of the ratio between Mtop and Mup(Bar) is. Setting S4 is used, but the width of Mtop is varied from $200nm$ to $3000nm$. Figure 4.15 shows that there is an obvious impact on delay and energy consumption. The impact on $\sigma_{V_{offset}}$ is more subtle. For very small widths, the effective V_{gs} of Mup and MupBar decreases, resulting in a slightly larger voltage gain before the NMOS transistors are enabled. The impact of V_T mismatch in Mup(Bar) increases, the impact of V_T mismatch in Mdown(Bar)

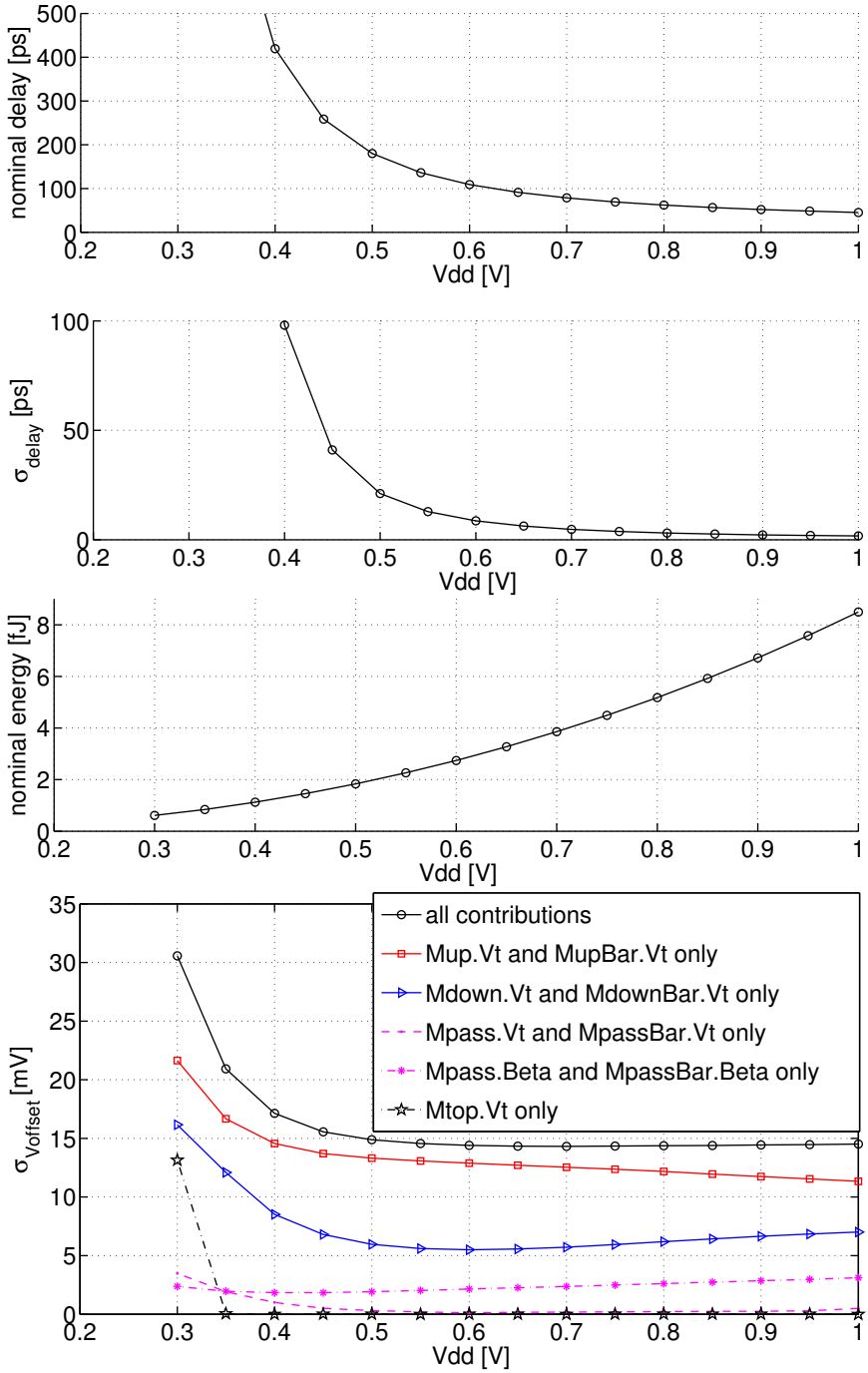


Figure 4.13: Behaviour of drain-input latch-type SA (setting S4, all SPLVT) as function of the supply voltage.

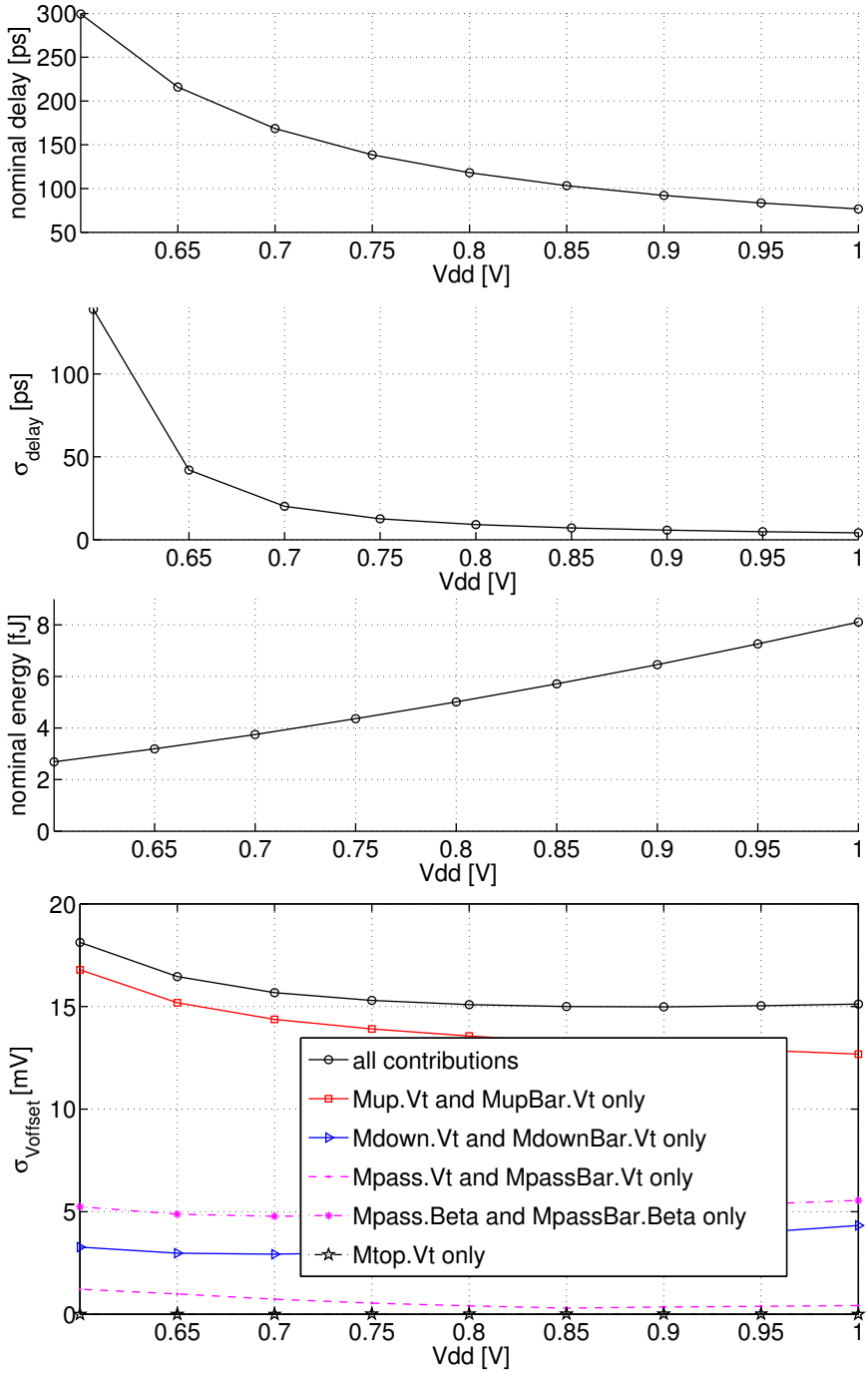


Figure 4.14: Behaviour of drain-input latch-type SA (setting S4, all SPHVT) as function of the supply voltage.

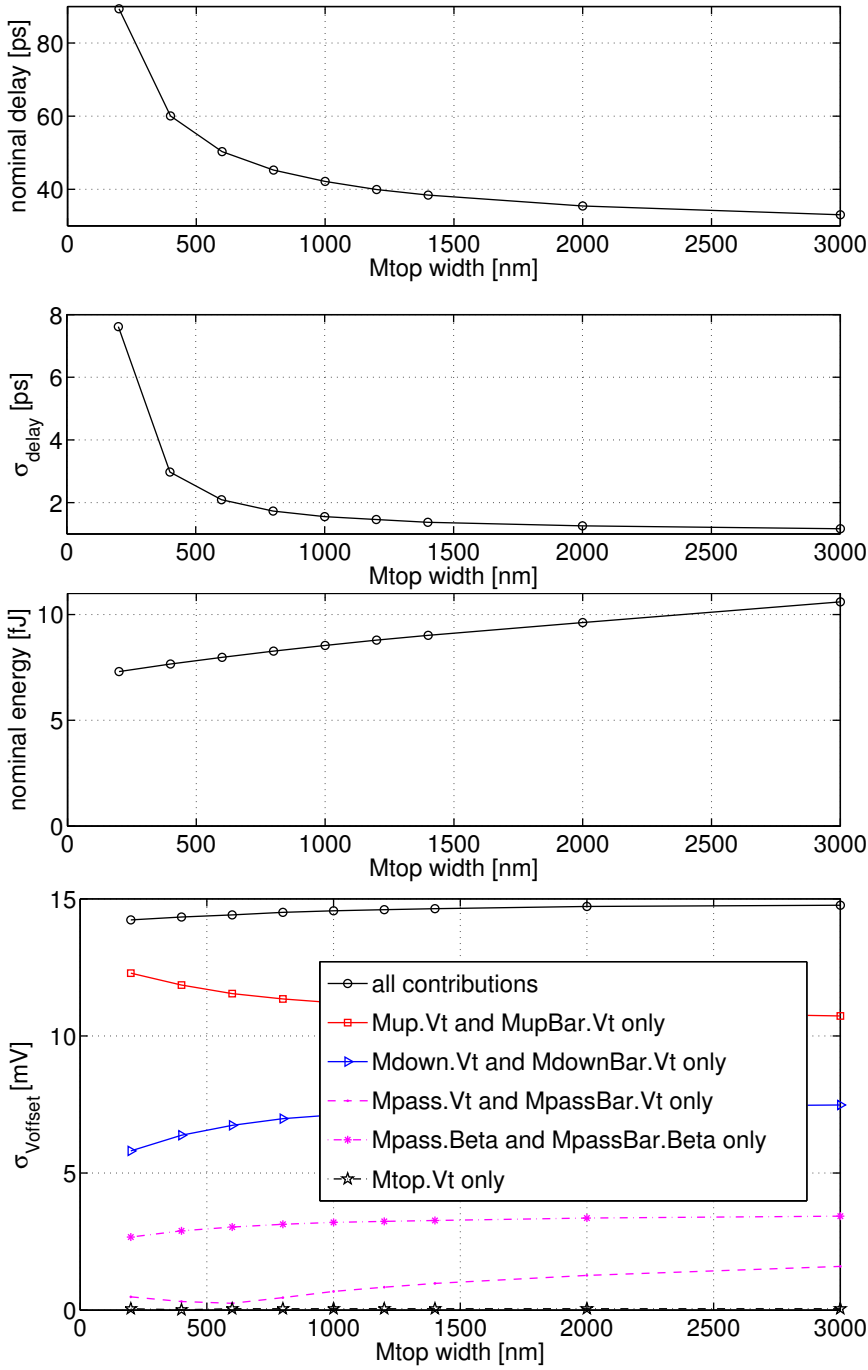


Figure 4.15: Behaviour of the drain-input latch-type SA with setting S4 as function of the width of Mtop.

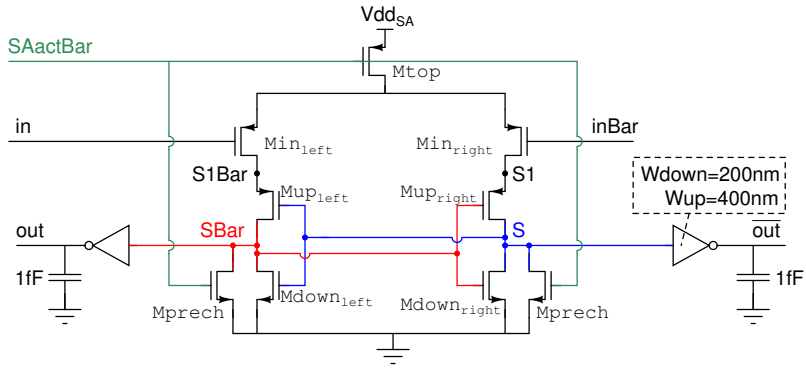


Figure 4.16: The schematic of the gate-input stacked latch-type SA. All transistors have minimal length ($L = 80nm$).

decreases. The net result is a slight decrease in $\sigma_{V_{offset}}$ from $14.8mV$ to $14.2mV$. This is probably not enough to accept the associated delay penalty.

4.6.2 Gate-input stacked latch-type SA

Figure 4.16 shows another often used strobed SA, the gate-input stacked latch-type SA. Conceptually, it consists of a differential pair Min(Bar) with switched current source Mtop. A (current-input) latch is stacked underneath to provide fast positive feedback and to stop static current consumption. The circuit was first presented in [Kob92], where it was called a current-mode latch SA as it was derived from a current-mirror SA.

Table 4.5 provides an overview of the different settings that were used in this section.

Figure 4.17 provides the nominal waveforms for the SA with setting S1. Notice the parasitic coupling from the input to node $S1Bar$. At the time the SA is activated, there is a voltage difference of $18mV$ between $S1$ and $S1Bar$ opposing the input signal. This effect is best seen on the graph of $\Delta S1$. This effect could be eliminated by properly equalising or precharging $S1$ and $S1Bar$ When the SA is activated, node *top* is pulled up, and transistors Min_{left} and $\text{Min}_{\text{right}}$ act as a differential amplifier. Node $S1Bar$ is pulled up faster than $S1$ because in is lower than $inBar$. At this time, signal $\Delta S1$ develops in the correct direction. When $S1Bar$ becomes large enough for Mup_{left} to conduct current, $SBar$ is pulled up. However, charging $SBar$ discharges $S1Bar$, and signal $\Delta S1$ develops in the wrong direction again while ΔS changes in the desired direction. When $SBar$ becomes large enough to enable $\text{Mdown}_{\text{right}}$, S is actively kept low.

At each point in time, the signal information is distributed between the input nodes, $\Delta S1$ and ΔS. The immediate change in ΔS is dictated by the difference in gate-source voltage between Mup_{left} and $\text{Mup}_{\text{right}}$, $\Delta V_{gs} = \Delta S1 + \Delta S$.

The internal operation of this SA is complicated and awkward. The stack of

setting	S1	S2	S3	S4	S5
Mtop W [nm]	800	800	1080	1080	1300
Min W [nm]	1080	1080	2700	2700	3200
Mup W [nm]	1080	1080	2700	2700	3200
Mdown W [nm]	600	600	2000	2000	2000
Mprech W [nm]	200	200	200	200	200
$V_{in,CM}$ [mV]	50	450	50	50	50
precharge $S1$ and $S1Bar$?	no	no	no	yes	yes
nominal delay [ps][a]	83	107	99.5	103	98
σ_{delay} [ps]	6.2	5.7	3.2	3.0	2.6
nominal energy [fJ]	14.4	12.3	29	31	35.3
τ_{10} $[\frac{ps}{decade}]$	25	40	22.5	22.5	25
$6.1 \cdot \sigma_{V_{offset}}$ [mV]	196	100	112	99.4	95
$\sigma_{V_{offset}}$ [mV]	32.2	16.4	18.4	16.3	15.5
contribution to $\sigma_{V_{offset}}$					
by Mup.Vt [mV]	13.3	1.7	7.6	7.3	6.7
by Mdown.Vt [mV]	14.4	4.8	6.6	5.9	5.9
by Min.Vt [mV]	11.6	9.9	7.1	5.6	5.2

Table 4.5: Overview table of different SA settings that were used for the gate-input stacked latch-type SA. Reported performance is at $Vdd = 1V$. All transistors are SPLVT, unless specified otherwise.

[a] All nominal values are simulated at $V_{in} = 100mV$

device	param	S' $[\frac{mV}{\sigma}]$	contribution	Width [nm]	Length [nm]	σ_i
$\text{Mup}_{\text{right}}$	Vt	13.3	41%	1080	80	8.9mV
Mup_{left}	Vt	9.6	30%	1080	80	8.9mV
$\text{Mdown}_{\text{right}}$	Vt	14.1	44%	600	80	13.2mV
$\text{Mdown}_{\text{left}}$	Vt	10.5	33%	600	80	13.2mV
$\text{Min}_{\text{right}}$	Vt	11.6	36%	1080	80	8.9mV
Min_{left}	Vt	11.6	36%	1080	80	8.9mV
	$\sigma_{V_{offset}}$:	$32.2mV$	100%			

Table 4.6: Overview table of the linear sensitivity analysis for V_{offset} for the gate-input stacked latch-type SA with sizing S1.

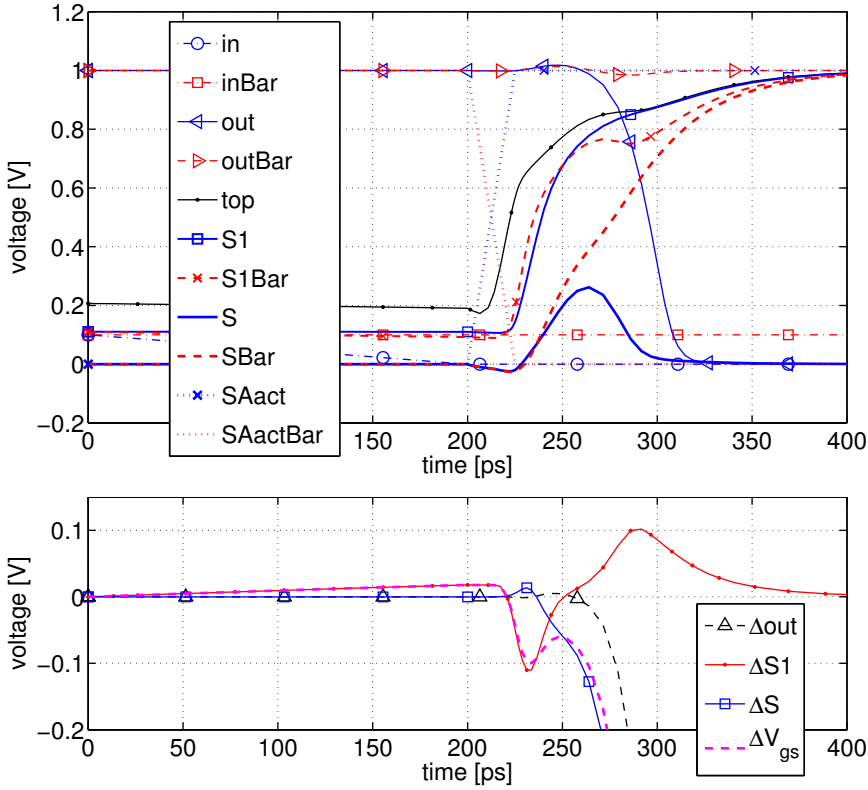


Figure 4.17: The nominal waveforms for the gate-input stacked latch-type SA with settings S1.

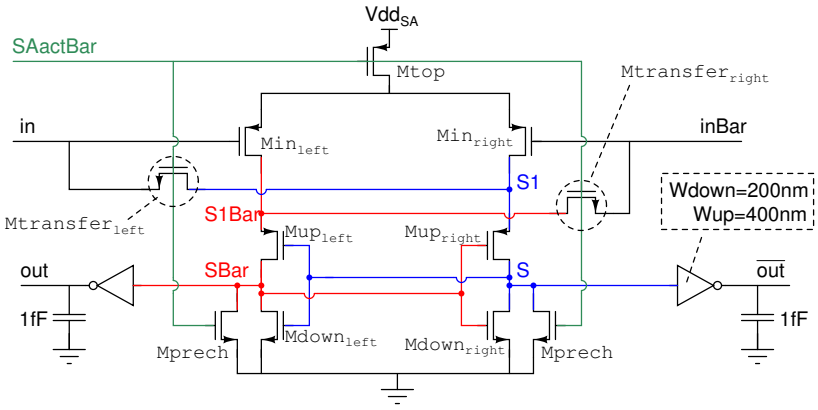


Figure 4.18: The schematic of the look-ahead SA. All transistors have minimal length ($L = 80nm$).

PMOS transistors results in slow, energy hungry operation, and the contention between signal on $S1$ and S results in a dramatically bad $\sigma_{V_{offset}}$, as can be seen from table 4.6. Many transistor threshold offsets are amplified rather than attenuated. This situation can be improved by increasing the common mode input voltage, as in setting S2. For our memory, this is not a valid option though. A similar yet smaller effect can be obtained by reducing the ratio of Mtop to Min, as in setting S3. Precharging the internal nodes $S1$ and $S1Bar$ to $0V$ provides a limited improvement in offset voltage, as shown by setting S4. The detrimental capacitive coupling from the inputs is avoided, and the input transistors have a little bit more voltage headroom. Setting S5 will be used for comparison later on.

This SA does have one advantage compared to the drain-input latch-type SA: it does not draw DC current from its inputs. This can be useful when a reference voltage is applied to one of the inputs, as it relaxes the specifications for the reference voltage generator. Notice that the delay of this SA should not be directly compared to the delay of the drain-input latch-type SA, as this SA does take late changes to the input signals into account, which probably makes a difference of about $20ps$.

4.6.3 Look-ahead type SA

In [Asa05], a look-ahead type SA was introduced. It is a variation on the gate-input latch-type SA, but the inputs are copied to the internal nodes $S1$ and $S1Bar$ before SA activation. This also resolves the problem with capacitive coupling. Figure 4.18 provides a schematic and table 4.7 provides an overview of the discussed settings.

Figure 4.19 shows the nominal waveforms for the circuit. Before the SA is activated, the input signal is indeed copied to the internal nodes $S1$ and $S1Bar$. The remaining SA operation is identical. The waveforms look less tricky than in

setting	S1	S2	S3
Mtop W [nm]	1080	1080	1080
Min W [nm]	1080	1080	2700
Mup W [nm]	1080	1080	2700
Mdown W [nm]	200	800	2000
Mprech W [nm]	200	200	200
Mtransfer W [nm]	200	200	200
nominal delay [ps][a]	68	77	68
σ_{delay} [ps]	3.6	3.4	1.5
nominal energy [fJ]	15	17	35
τ_{10} $[\frac{ps}{decade}]$	26	27	18
$6.1 \cdot \sigma_{V_{offset}}$ [mV]	166	142	91.5
$\sigma_{V_{offset}}$ [mV]	27.3	23.3	15.0
contribution to $\sigma_{V_{offset}}$			
by Mup(Bar).Vt [mV]	9.7	9.5	6.3
by Mdown(Bar).Vt [mV]	13.4	9.1	6.3
by Min(Bar).Vt [mV]	6.7	6.9	4.2

Table 4.7: Overview table of different SA settings that were used for the look-ahead SA. Reported performance is at $Vdd = 1V$. All transistors are SPLVT, unless specified otherwise.

[a] All nominal values are simulated at $V_{in} = 100mV$

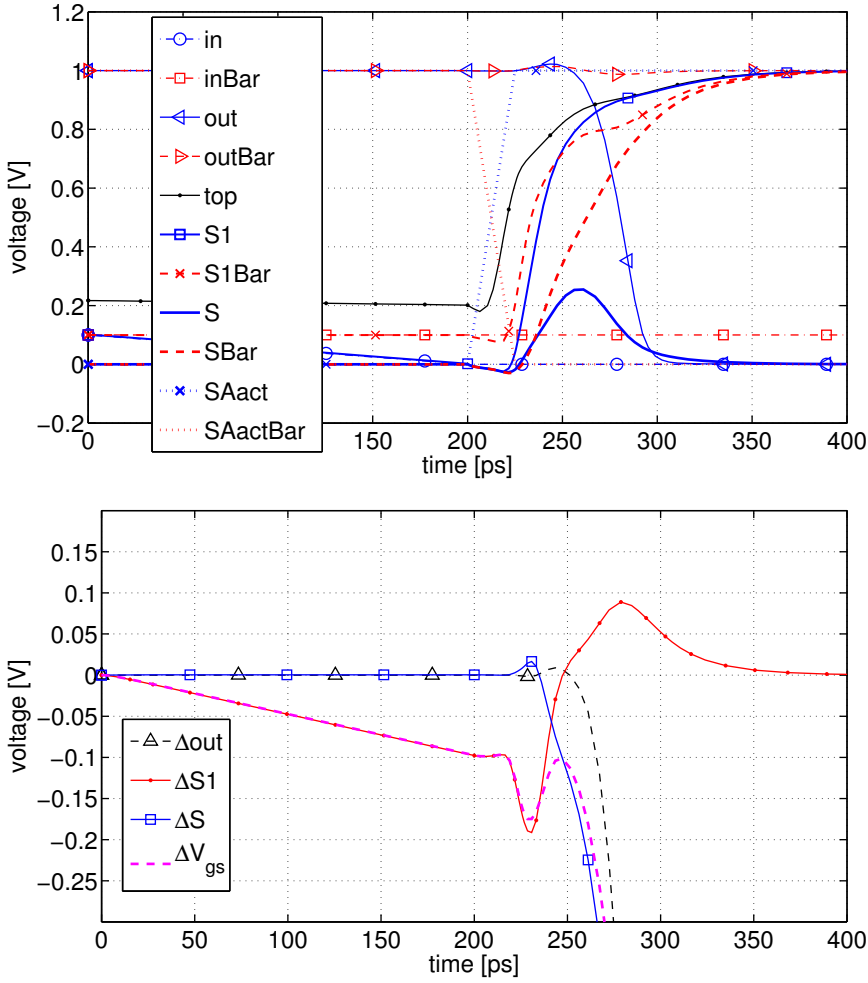


Figure 4.19: The nominal waveforms for the look-ahead SA with settings S1.

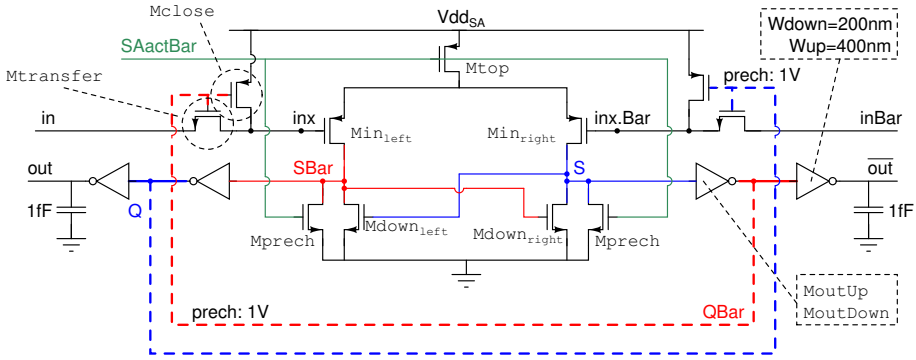


Figure 4.20: The schematic of the self-closing SA. All transistors have minimal length ($L = 80nm$).

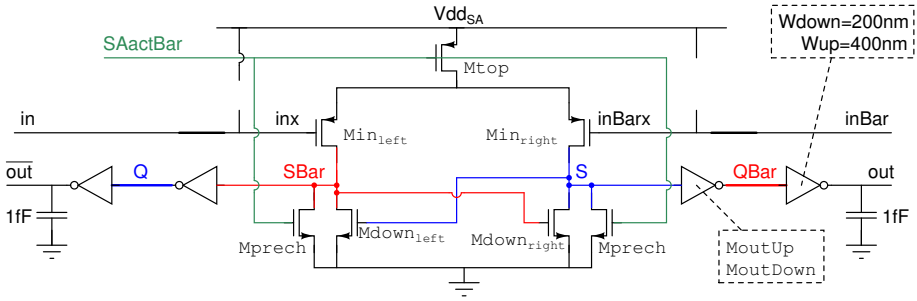


Figure 4.21: The schematic of the core part of the self-closing SA.

the gate-input latch-type SA. The offset voltage in setting S1 and S2 indicates that this SA topology is still very sensitive to transistor sizing. When setting S3 of the look-ahead SA is compared to setting S5 of the gate-input stacked latch-type SA, the main advantage seems to be a significant reduction in delay for a similar energy consumption and offset.

4.6.4 Self-closing SA (design 1)

Figure 4.20 shows the schematic of the self-closing SA that was used in our $0.18\mu m$ design [Cos07]. The main goal was to avoid the stack of transistors in the gate-input stacked latch-type SA, as this stack requires too much voltage headroom and introduces too many transistors that are critical for offset and speed. Figure 4.21 shows the core part of the self-closing SA. Because Q and QBar are at $1V$ during precharge, the self-closing SA indeed resembles this simpler setup during the most important phase of the sensing operation. The additional transistors Mtransfer and Mclose are added to stop the static current flow when sensing is complete. Table 4.7 gives an overview of specifications for some different sizings.

setting	S1	S2	S3	S4	S5
top W [nm]	800	200	800	600	600
Min W [nm]	1080	1080	1080	1200	1800
Mdown W [nm]	600	600	1080	800	1200
Mprech W [nm]	200	200	200	200	200
Mtransfer W [nm]	200	200	200	200	200
Mclose W [nm]	200	200	200	200	200
MoutDown W [nm]	400	400	400	400	400
MoutUp W [nm]	200	200	200	200	200
nominal delay [ps][a]	54	106	59	63	71
σ_{delay} [ps]	2.2	9.4	2.5	2.8	3.3
nominal energy [fJ]	16.6	13.5	18	16.6	20
τ_{10} $[\frac{ps}{decade}]$	28	27.5	19.2	22	21
$6.1 \cdot \sigma_{V_{offset}}$ [mV]	128	99	118	110	88
$\sigma_{V_{offset}}$ [mV]	21	16.3	19.4	18	14.5
contribution to $\sigma_{V_{offset}}$					
by Min.Vt [mV]	8.7	8.9	8.7	8.3	6.8
by Mdown.Vt [mV]	9.2	6.6	7.9	7.7	5.7

Table 4.8: Overview table of different SA settings that were used for the self-closing SA. Reported performance is at $Vdd = 1V$. All transistors are SPLVT, unless specified otherwise.

[a] All nominal values are simulated at $V_{in} = 100mV$

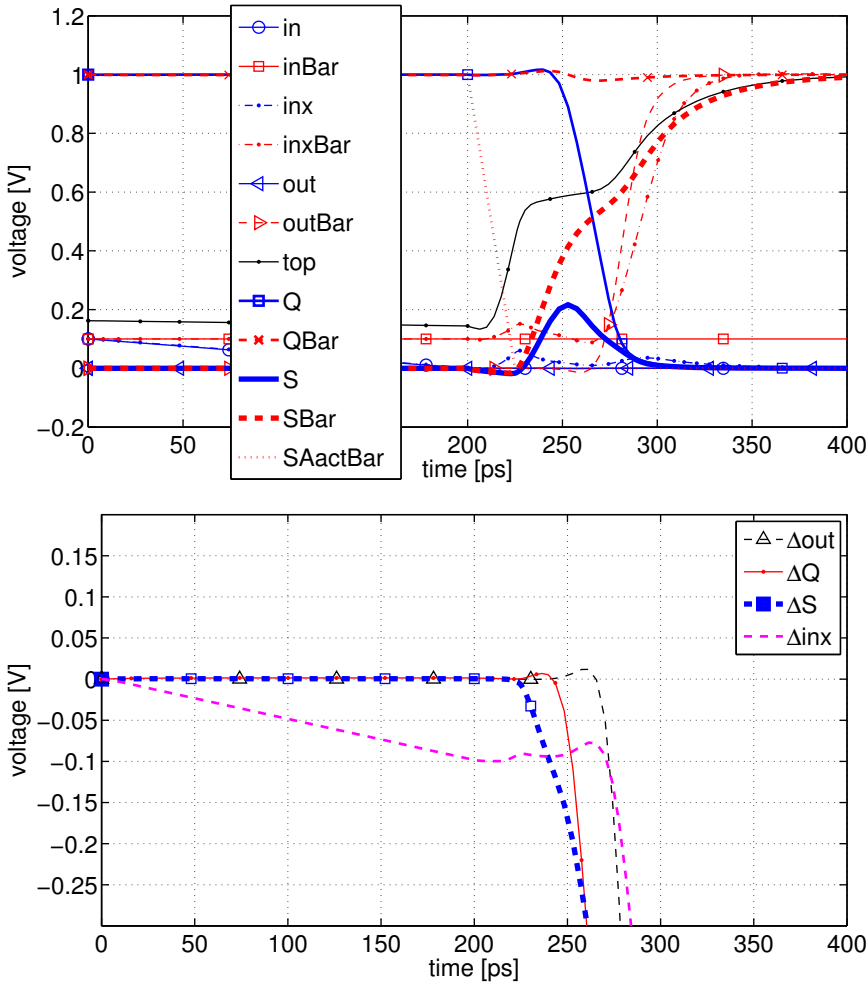


Figure 4.22: The nominal waveforms for the self-closing SA with settings S5.

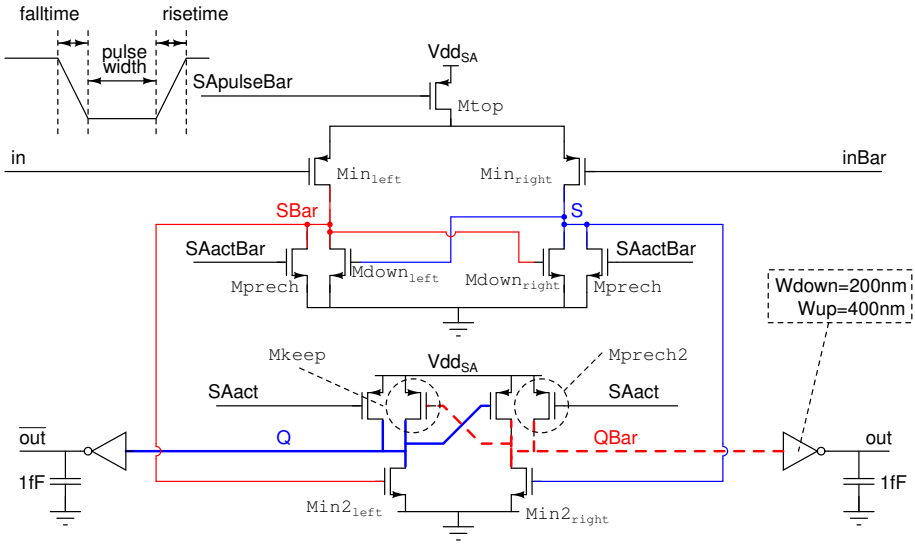


Figure 4.23: The schematic of the pulsed current source SA.

Figure 4.22 shows the nominal waveforms for the self-closing SA. Before SA activation, inx tracks in. This tracking continues until sensing is almost finished. When Mtop is enabled, the input transistors Min act as a differential pair. $SBar$ increases faster than S because inx is lower than $inxBar$. When $SBar$ becomes higher than the NMOS threshold voltage, S is pulled down by $\text{Mdown}_{\text{right}}$. At about the same time, node Q is pulled down. As a consequence, $outBar$ goes high. When Q goes low, the transfer transistor $\text{Mtransfer}_{\text{right}}$ is disabled, and $\text{Mclose}_{\text{right}}$ is enabled, which pulls node $inxBar$ to Vdd. This stops the static current that was flowing through Mtop, $\text{Min}_{\text{right}}$ and $\text{Mdown}_{\text{right}}$. When the SA gets disabled, charge flows from $inxBar$ to the input.

The introduction of Mtransfer results in some signal loss: input signal tracking before activation is good but not perfect, and there is some differential impact of the cross-coupled capacitance at the onset of sensing. Still, the energy consumption of the self-closing SA is significantly lower than that of the look-ahead SA for a similar ΔV_{min} and delay.

4.6.5 Pulsed current source SA (design 2)

Figure 4.23 shows the schematic of the pulsed current source SA that was used in our 90nm design [Cos08]. It is an improvement of the self-closing SA. All current that flows in the first stage of the self-closing SA after the point where the output inverters start to toggle, is actually a waste of energy. The circuitry to close the SA consumes quite some energy as well. All this is avoided by moving to a pulsed SA scheme. The concept of pulsed SA operation is not new in itself: it was already mentioned in [Kob92] for current-mode SAs.

setting	S1	S2	S3	S4
top W [nm]	600	600	600	600
Min W [nm]	1500	1500	1500	1500
Mdown W [nm]	1000	1000	1000	1000
Mprech1 W [nm]	200	200	200	200
Mkeep W [nm]	200	200	200	200
Mprech2 W [nm]	200	200	200	200
Min2 W [nm]	400	400	400	400
pulse risetime [ps]	25	25	25	25
pulse peak duration [ps]	100	50	25	10
pulse falltime [ps]	25	25	25	25
nominal delay [ps][a]	62.0	62.1	62.9	75
σ_{delay} [ps]	2.8	2.8	4.4	16.8
nominal energy [fJ]	19.6	13.7	10.6	8.6
τ_{10} $[\frac{ps}{decade}]$[b]	15	15	15	30
required $\Delta V_{min,overdrive}$ for nominal delay of $100ps$ [mV]	0.5	0.5	3.9	36
$6.1 \cdot \sigma_{V_{offset}}$ [mV]	91	91	89	88
$\sigma_{V_{offset}}$ [mV]	14.92	14.85	14.63	14.48
contribution to $\sigma_{V_{offset}}$				
by Min.Vt [mV]	7.6	7.6	7.6	7.6
by Mdown.Vt [mV]	6.2	6.2	6.1	6.1
...	...	...	...	...
by Min2.Vt [mV]	1.2	1.2	1.7	2.8

Table 4.9: Overview table of different SA settings that were used for the pulsed current source SA. Reported performance is at $Vdd = 1V$. All transistors are SPLVT and have minimal length ($L = 80nm$), unless specified otherwise.

[a] All nominal values are simulated at $V_{in} = 100mV$

[b] Measured on the V_{in} interval where the logarithmic rule is valid

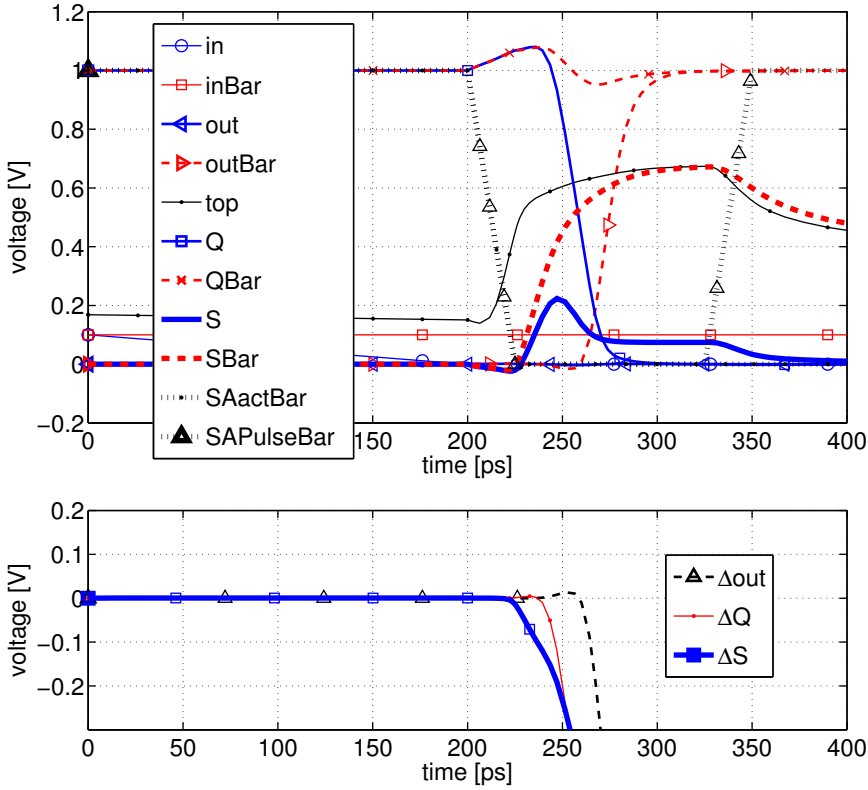


Figure 4.24: The nominal waveforms for the pulsed current source SA with settings S1.

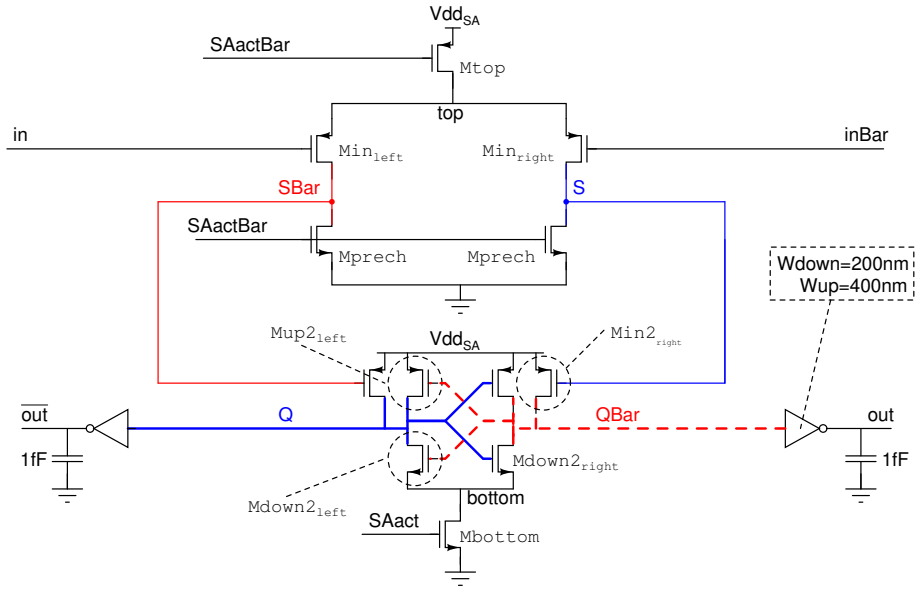


Figure 4.25: The schematic of the double-tail latch-type SA.

Table 4.9 shows the different settings that have been tested for this SA. The only difference is in the duration of the pulse width of $SApulseBar$. The SA operates reliable with pulse widths between $100ps$ and $25ps$. In setting S4, with a pulse width of $10ps$, the delay goes up steeply under variations. If timing can be controlled to be between $25ps$ and $50ps$, this SA is both faster and less energy hungry than the self-closing SA. The input signals for this SA arrive at a gate input, so they do not have to deliver DC current, which was important in our design.

4.6.6 Double-tail latch-type SA

In [Sch07], the double-tail latch-type SA of figure 4.25 was introduced. The authors state that the double tail allows for a separate optimisation of the two stages. The first stage uses a small current source to minimize the offset. The second stage uses a large current for fast latching.

Figure 4.26 shows the waveforms for a nominal access with $100mV$ input swing. The first stage is rather straightforward. Nodes S and $SBar$ are precharged to ground. When the SA is enabled, the PMOS input transistors pull both nodes up. $SBar$ is pulled up a little faster than S. ΔS increases. When the input transistors enter the linear region, the voltage difference ΔS is destroyed again.

The second stage is much more tricky. Because S and $SBar$ are precharged to ground, Q and $QBar$ are precharged to Vdd by $\text{Min2}_{\text{left}}$ and $\text{Min2}_{\text{right}}$. On SA activation, the NMOS transistors of the cross-coupled latch try to amplify ΔQ. Initially, ΔQ is not related to the input signal, hence transistors Min2

setting	S1a	S1b	S2
$V_{in,CM}$ [V]	1.1	0.7	1.1
nominal delay [ps][a]	55	62	t0
nominal energy [fJ]	40	40	110
τ_{10} $[\frac{ps}{decade}]$	37	37	44
$6.1 \cdot \sigma_{V_{offset}}$ [mV]	79	91	49
$\sigma_{V_{offset}}$ [mV]	13	13	8
$\sigma_{v_{noise}}$ [mV]	N.A.	N.A.	1.5

Table 4.10: Result table for the double-tail latch-type SA as reported in [Sch07]. They used a $1.2V$ $90nm$ technology. NMOS an PMOS stage are exchanged compared to figure 4.25. A common mode input between $1.1V$ and $0.7V$ is supported. Transistor sizings an threshold voltages were not disclosed.

[a] At $V_{in} = 100mV$

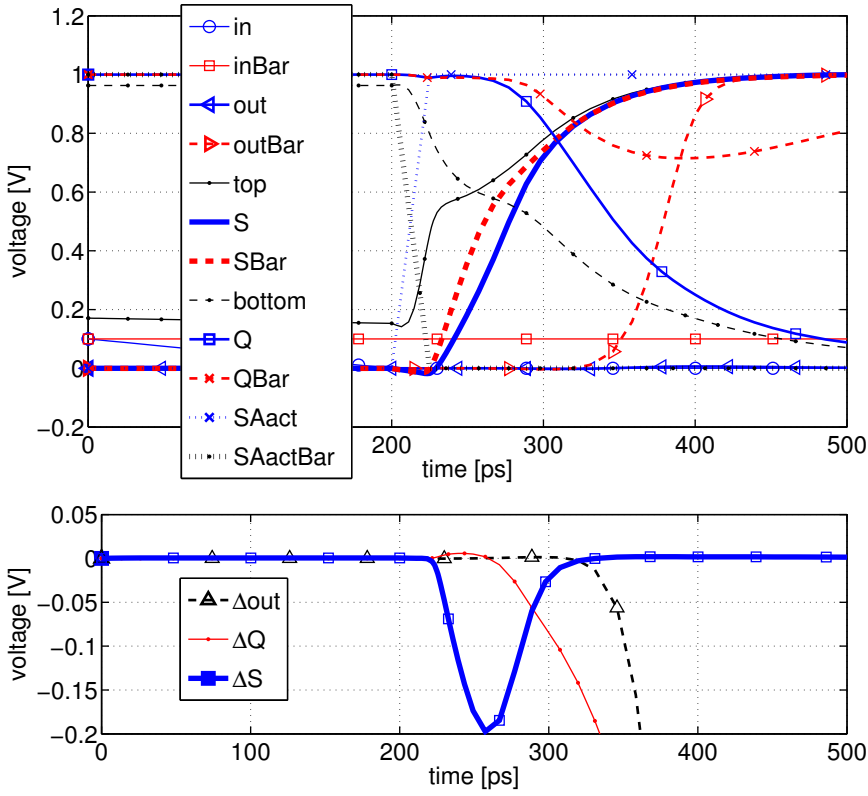


Figure 4.26: Example waveforms for the double-tail latch-type SA from figure 4.25.

drain-input latch-type SA setting	S1a	S1b
Mtop W [nm]	800	$4 \cdot 800$
Mup W [nm]	1080	$4 \cdot 1080$
Mdown W [nm]	600	$4 \cdot 600$
Mbottom W [nm]	200	$4 \cdot 200$
Mpass W [nm]	200	$4 \cdot 200$
estimated C_0[a] [fF]	3.4	13.6
estimated A_{pre}	1	1
simulated $\sigma_{V_{offset}}$ [mV]	1.45	0.73
estimated $\sigma_{V_{offset}}$ $\frac{1}{A_{pre}}\sqrt{2}\sqrt{\frac{kT}{C_0}}$ [mV]	1.56	0.78

Table 4.11: Results of transient noise simulations for drain-input latch-type SA. All transistors are SPLVT and have minimal length ($L = 80nm$), unless specified otherwise.

[a] As reported by spectre's info-statement

need to fix ΔQ to 0 until a sufficient signal has been developed on S and $SBar$. Otherwise, noise and offset will be amplified instead of the input signal. $SBar$ increases faster than S, so $\text{Min2}_{\text{left}}$ will become weaker than $\text{Mdown2}_{\text{left}}$ first. When this happens, positive feedback is triggered.

This operation has two awkward features. Firstly, the first stage has only a limited time period in which ΔS has a useful value. Secondly, The sizing of Mdown2, the switched current source of the second stage, must avoid two problematic regions. If Mdown2 is too weak, signal ΔQ can not develop because Q and $QBar$ stay near Vdd. If Mdown2 is too strong before enough signal ΔQ was build up, the cross-coupled latch tries to amplify an attenuated version of the input signal, which would make the second stage dominant for mismatch. This effect is made even worse by the capacitive coupling of S to $QBar$ and $SBar$ to Q, which forces ΔQ in the wrong direction at first.

Table 4.10 provides the performance numbers that were reported in [Sch07]. They also used a $90nm$ technology, so performance numbers of settings S1a and S1b can more or less be compared with the other discussed SAs. This topology is not the best choice for low-power memories.

4.6.7 Noise performance

Noise performance was simulated for the drain-input latch-type SA and for the pulsed current source SA. The results are summarised in tables 4.11 and 4.12. The simulations were performed without output load. The simulated values for $\sigma_{V_{offset}}$ are obtained as explained in section A.

pulsed current source SA setting	S2a	S2b	S2c
Mtop W [nm]	600	$4 \cdot 600$	600
Min W [nm]	1500	$4 \cdot 1500$	1500
Mdown W [nm]	1000	$4 \cdot 1000$	1000 (HVT)
Mprech1 W [nm]	200	$4 \cdot 200$	200
Mkeep W [nm]	200	$4 \cdot 200$	200
Mprech2 W [nm]	200	$4 \cdot 200$	200
Min2 W [nm]	400	$4 \cdot 400$	400
$V_{in,CM}$ [mV]	0	0	500
estimated C_0 [fF]	4.1	16.4	4.1
estimated A_{pre}	1.4	1.4	2.6
simulated $\sigma_{V_{offset}}$ [mV]	0.85	0.45	0.53
estimated $\sigma_{V_{offset}}$ $\frac{1}{A_{pre}}\sqrt{2}\sqrt{\frac{kT}{C_0}}$ [mV]	1.01	0.51	0.55

Table 4.12: Results of transient noise simulations for pulsed current source SA. All transistors are SPLVT and have minimal length ($L = 80nm$), unless specified otherwise.

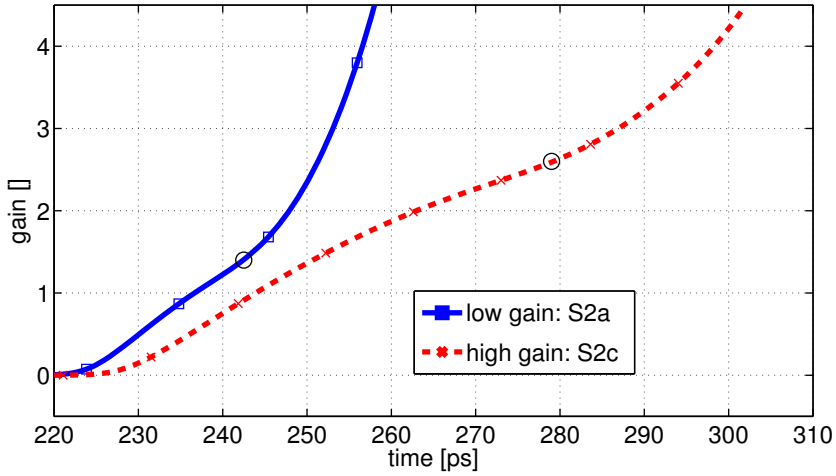


Figure 4.27: The amplification $A = \frac{v_{out}}{v_{in}}$ as function of time for a small input voltage for two setups from table 4.12. The points where positive feedback starts to dominate are indicated with circles.

	$\sigma_{V_{offset}}$ [mV]	energy [fJ]	delay[a] [ps]
drain-input latch-type	14.5	8.3	45
gate-input stacked-latch type	15.5	35.3	98
look-ahead	15.0	35	68
self-closing	14.5	20	71
pulsed current source	14.9	13.7	62
double-tail latch-type	13	40	55

Table 4.13: Performance values for different SA types at $Vdd = 1V$.

[a] At $V_{in} = 100mV$

The results are described rather well by this simple formula:

$$\sigma_{V_{offset}} \approx \frac{1}{A_{pre}} \sqrt{2} \sqrt{\frac{kT}{C_0}} \tag{4.4}$$

C_0 is the capacitance on the internal SA nodes that are part of the positive feedback loop. The value was obtained at the onset of SA activation with the captab option of spectre's info statement. A_{pre} is the amplification from the input voltage to the sensitive nodes just before positive feedback begins to dominate the SA operation. For the drain-input latch-type SA, this is at most 1 because all amplification is performed by positive feedback. For the pulsed current source SA, A_{pre} can be estimated by looking at $\Delta S = S - SBar$ when a very small input voltage is applied. Figure 4.27 shows $\frac{\Delta S}{v_{in}}$. At first, the slope is linear. When positive feedback becomes important, the slope increases exponentially over time.

This allows for a simple interpretation of the impact of thermal noise on the comparator: at the onset of positive feedback, the noise voltage difference between the internal nodes is sampled and amplified. Each sampled noise voltage is a sample of an independent Gaussian distribution with $\sigma_{v_{n,internal}} = \sqrt{\frac{kT}{C_0}}$. The voltage difference between the two nodes is again a Gaussian distribution with a $\sqrt{2}$ times larger standard deviation. To balance this sampled noise voltage, an input voltage v_{in} is needed. v_{in} is amplified by a factor A_{pre} before the sampling occurs, so it is $v_{in}A_{pre}$ which is compared to the sampled noise voltage difference.

4.6.8 Comparison

Table 4.13 combines the results for the different SA types. To be able to compare them quantitatively, each of the SAs is designed for the same specifications: an input signal $\Delta V_{min} = 100mV$, a SA failure rate $fr = 10^{-9}$ ($n = 6.1$) and a delay below $100ps$.

The drain-input latch-type SA requires less energy than the other solutions and has a very competitive delay. The only two drawbacks of this SA are its noise

performance ($A_{pre} = 1$) and the fact that the inputs need to be able to deliver net charge. This last disadvantage was problematic for the $90nm$ design[Cos08], in which reference voltages were generated by a simple resistive divider. In all use cases where these two drawbacks are not unsurmountable, the drain-input latch-type SA should be used.

4.7 Summary

This chapter provides an in-depth analysis of strobed voltage-mode sense amplifiers. Section 4.2 introduces the concept of the SA offset voltage V_{offset} and discusses the most important "noise" sources that negatively affect the SA operation, including transistor mismatch and thermal noise. Section 4.3 describes the *ecosystem* in which the SA operate.

Section 4.4 discusses the important specifications for SAs: the required differential input signal ΔV_{min}, the energy consumption per operation E_{SA} and the SA delay.

Section 4.5 outlines a practical simulation-based analysis method for strobed voltage mode SAs, including the impact of transistor mismatch and thermal noise.

Section 4.6 evaluates several implementations of strobed voltage mode SAs. A simple equation to estimate the noise performance of strobed SAs with regenerative feedback is validated against simulation results (equation 4.4).

The drain-input latch-type sense amplifier is clearly the best sense amplifier in the comparison. Only when noise performance or input current is a dominant problem, the pulsed current source SA which we re-introduced is the best alternative. In our example 90nm technology, the input transistors of a drain-input latch-type SA must be $1080nm$ wide (9x $120nm$) to be able to sense a $100mV$ complementary input signal. With correct sizing, SA offset is dominated by the input transistors. This SA has a dynamic energy consumption of $8.3fJ$ at $1V$. Figure 4.13 shows that voltage scaling down to $450mV$ does not introduce problems when SPLVT transistors are used. At $450mV$, the dynamic energy consumption is reduced from $8.3fJ$ to $1.44fJ$.

Contributions and highlights

The analysis in this chapter shows that many of the sense amplifiers that are used in today's memories behave poorly under intra-die variations.

This work is the first to highlight the growing importance of thermal noise in memory SAs. It shows that the effects of thermal noise can be mitigated by using a SA with a pre-amplification A_{pre} before positive feedback kicks in. Of the SAs with pre-amplification, the pulsed current source SA that is used in this work provides the best trade-off between SA energy and offset voltage. An additional advantage of this SA is that it does not draw net current from its inputs.

Chapter 5

Sense Amplifier Calibration

5.1 Introduction

In the previous chapter, the design of traditional strobed voltage-mode SAs was discussed. For these traditional SAs, Pelgrom's law puts a fundamental limitation on the SA performance, as discussed in section 4.4.1.1.

Section 5.2 recapitulates these limitations. This section also discusses why it is important to overcome these limitations, especially for memories with a large word length and for memories in DDSM technologies.

This chapter discusses SA calibration, a very effective technique to overcome the traditional limitations on SA performance. Sense amplifier calibration is a family of techniques that solves the SA offset problem without increasing the size of the SA transistors. Calibration is performed in a separate calibration phase, for example after fabrication, while booting or whenever temperature has changed by 5 degrees. During this phase, the best setting for each component is obtained and stored. During normal accesses, these settings can be used without much energy and delay overhead and without static current consumption.

Section 5.3 treats SA redundancy, a type of SA calibration introduced in [Ver08b]. Here, each SA is replaced with N smaller SAs, of which the best one is selected. Section 5.4 discusses SA tuning, a novel type of SA calibration that we introduced [Cos08, Cos09]. The concept, an example implementation and a way to calibrate the tunable sense amplifiers are presented. Section 5.5 introduces an interesting novel variation on SA redundancy. With asymmetric SA redundancy, rather than providing N equal-sized smaller SAs, The SA sizes are selected as to minimize the average SA energy consumption. Section 5.6 provides a comparison of the different calibration techniques and summarizes this chapter.

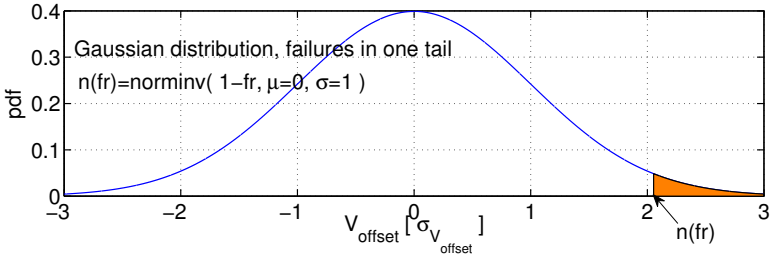
5.2 Limitations of traditional SA designs

For traditional SAs, Pelgrom's law puts a fundamental limitation on the SA performance, as discussed in section 4.4.1.1.

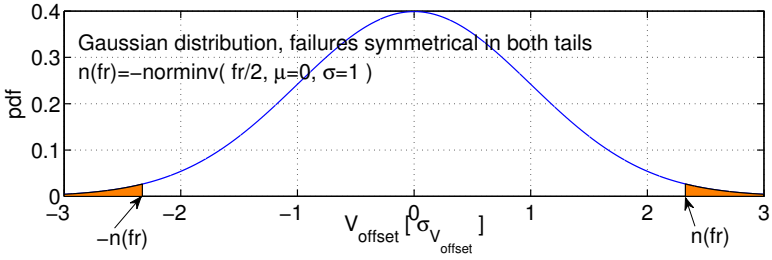
$$\Delta V_{min} \geq n(fr)\sigma_{V_{offset}} \tag{5.1}$$

$$= n(fr)\frac{A_{\Delta V_T}}{\sqrt{W \cdot L}} \tag{5.2}$$

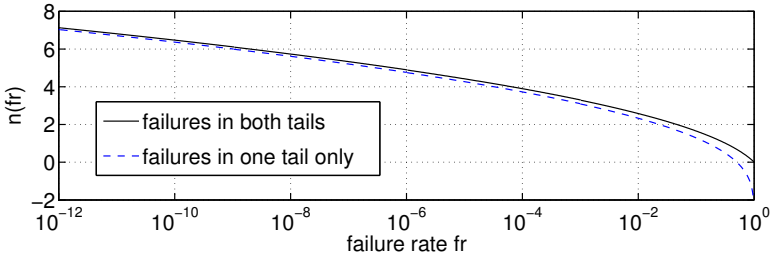
$$= n(fr)\frac{\sigma_{\Delta V_{T,min}}}{\sqrt{S}} \tag{5.3}$$



(a) Only failures in one tail.



(b) Failures in both tails.



(c) $n(fr)$ as function of fr

Figure 5.1: $n(fr)$ indicates the range of correct operation on the x-axis of the pdf. For SA V_{offset}, errors occur in both tails.

ΔV_{min} is the minimal input voltage difference between the SA inputs that is required to obtain the desired failure rate fr. In a setup with complementary bit lines, V_{swing} must be at least ΔV_{min}. In a setup with single-ended signalling, V_{swing} must be at least $2 \cdot \Delta V_{min}$. W and L are the dimensions of the input transistors. S is the upscale factor $S = \frac{W \cdot L}{W_{min} \cdot L_{min}}$. S is a good indicator of the relative area and dynamic energy consumption of SAs for a fixed SA topology.

$n(fr)$ multiplied by $\sigma_{V_{offset}}$ is the range on the x-axis of the probability distribution of V_{offset} for which sensing occurs correctly. This is graphically represented in figure 5.1. A SA fails both when its offset is too large and when its offset is too large, hence in this chapter $n(fr)$ is

$$n(fr) = -norminv(fr/2, \mu = 0, \sigma = 1) \tag{5.4}$$

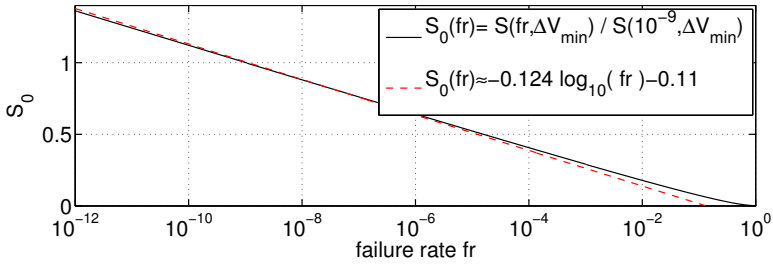


Figure 5.2: Relative size of a traditional SA as function of the allowed failure rate fr.

node	$A_{\Delta V_T}$ $[mV\mu m]$ (estimated)	required SA upscaling[a] and SA energy contribution[b] ΔV_{min}[c] $100mV$		ΔV_{min}[c] $50mV$		ΔV_{min}[c] $25mV$	
90nm	4.5	6.8x	(10%)	27x	(40%)	108x	(160%)
65nm	4	9.8x	(15%)	39x	(58%)	158x	(233%)
45nm	3.2[d]	12.6x	(19%)	50.4x	(75%)	202x	(299%)
32nm	2.8[d]	19.6x	(29%)	78.4x	(116%)	314x	(465%)

Table 5.1: Required size of sense amplifier input transistors to cope with SA offset due to transistor mismatch.

[a]required size for the SA input transistors (for 6σ) . [expressed in multiples of the technology's W_{min}, with $L = L_{min}$]
[b]SA energy consumption [fraction of consumption of total memory, SAs excluded]
[c]voltage difference between inputs. Single-ended SAs require twice this as BL swing.
[d]optimistic estimate, assumes high-K metal gate stack

Equation 5.3 can be rewritten as

$$S \geq \left(n(fr) \frac{\sigma_{\Delta V_{T,min}}}{\Delta V_{min}} \right)^2 \tag{5.5}$$

S is a function of fr and ΔV_{min}. In this chapter, we are interested in comparisons between different SA implementations for the same value of ΔV_{min}. We can hence use normalized upscale factors. $S_0(fr)$ is the normalized upscale factor for a traditional SA that results in a failure rate fr.

$$S_0(fr) = \frac{S(fr, \Delta V_{min})}{S(fr, 10^{-9})} = \left(\frac{n(fr)}{n(10^{-9})} \right)^2$$

Notice that $S_0(fr)$ is indeed independent of ΔV_{min}. Figure 5.2 shows $S_0(fr)$, along with the approximation $S_0(fr) \approx -0.124 \log_{10}(fr) - 0.11$, which is adequate when $fr < 10^{-3}$.

In traditional SA designs, smaller input signals can only be accommodated by increasing the area of the critical SA transistors to improve their matching. This

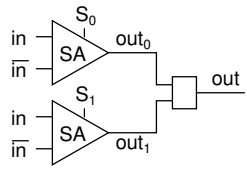


Figure 5.3: Setup for SA redundancy (N=2).

increases its dynamic energy consumption E_{SA}. In this chapter, we assume a simple relation between E_{SA} and the upscale factor S of the input transistors:

$$E_{SA} \sim S \tag{5.6}$$

This relation is a reasonable approximation as long as S is large. When S is small, the contribution of the non-critical transistors in the SA to E_{SA} is significant. These transistors cannot be reduced in size along with S as they are already minimal size.

Table 5.1 provides an example to clarify the implications of the relation that Pelgrom's law imposes on E_{SA} and ΔV_{min}. The example values are based on our 90nm design [Cos08, Cos09]. When a SA is sized to sense a differential input signal of $100mV$ in a 90nm technology, the area of the input transistors needs to be scaled up with a factor of 6.8x. More accurate numbers for strobed SAs can be found in chapter 4. If the pulsed current source SA is used with a $50ps$ pulse width, the required energy consumption that is needed to sense $100mV$ with a failure rate below 10^{-9} is $13.7fJ$ at $1V$ supply voltage. The total energy consumption per read access of our 90nm prototype memory in the 780MHz mode is $3880fJ$. The memory needs 32 SAs, so SA energy is about 10% of the total memory read energy. If a smaller ΔV_{min} is desired to reduce the energy consumption and delay on the bit lines, E_{SA} increases quadratically, as indicated by equation 5.5. A single-ended bit line with a swing of $100mV$ corresponds to $\Delta V_{min} = 50mV$. With a traditional SA, this would require $S = 27$x and a correspondingly large E_{SA}. The expected contribution of SA energy to the total read access energy of the memory is expected to increase to 29% in 32nm, even when optimistic values are assumed for Pelgrom's constant and the differential input signal is maintained at $100mV$.

Calibration techniques as discussed in this chapter can reduce the required input swing for a fixed SA energy consumption and delay with a factor of 8 and more. This certainly allows to improve the energy-delay performance in todays memories. Such Large reductions will prove especially useful in larger memories, in technologies with larger variations as well as in other applications.

5.3 SA redundancy

In [Ver08b], SA redundancy was introduced to improve SA yield for a given SA area. Fig. 5.3 illustrates the concept. Each SA is replaced with a set of N (≥ 2)

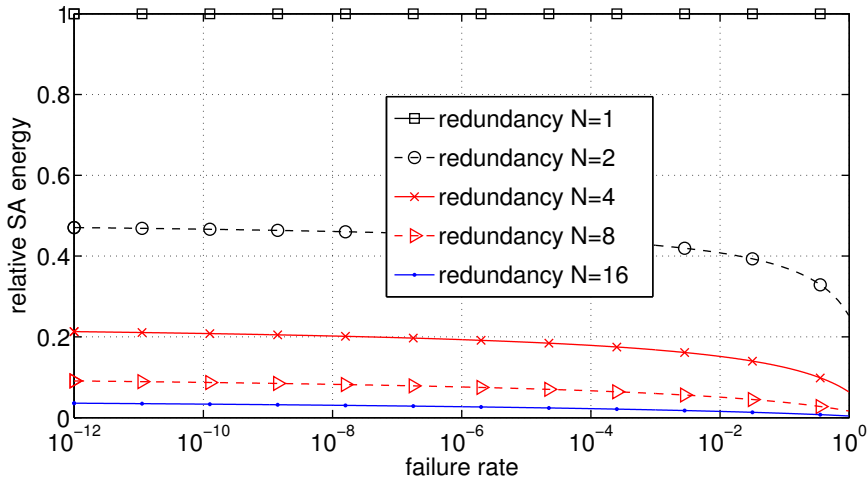


Figure 5.4: RS_N, the relative size of the individual SAs when N-fold SA redundancy is used. This ratio is a good measure for the energy reduction provided by SA redundancy for a fixed input signal. $N = 1$ is a traditional SA.

SAs. During calibration, the best SA of each set is determined. Afterwards, only the best SAs will be used. The resulting effective SA offset is the minimum of the offsets of the SAs in the set, which results in a much more favorable effective offset distribution. The actual selection of the signal from the best SA to the output must happen at run time. This introduces an overhead on access time and energy. This overhead increases as N increases, which limits this scheme to relatively small values of N.

With sense amplifier redundancy, a true SA failure only occurs if all N SAs fail. With $N = 2$ and under the assumption that SA failures are independent, the resulting failure rate after calibration, $fr_{redundancy}$, is

$$\begin{aligned} fr_{redundancy} &= fr = P[\text{SA}_1 \text{ fails } \& \text{ SA}_2 \text{ fails } \& \ldots \& \text{ SA}_N \text{ fails}] \\ &= P[\text{SA}_1 \text{ fails}] \cdot P[\text{SA}_2 \text{ fails}] \cdot \ldots \cdot P[\text{SA}_N \text{ fails}] \\ &= fr_1 \cdot fr_2 \cdot \ldots \cdot fr_N = fr_{native}^N \end{aligned} \tag{5.7}$$

To obtain a given failure rate fr, each of the individual SAs need a native failure rate fr_{native}

$$fr_{native} = \sqrt[N]{fr} \tag{5.8}$$

We are interested in the area and energy gain that is obtained with this relaxation of the failure rate. Without SA redundancy, the allowed failure rate fr puts the following constraint on SA size S, differential input signal ΔV_{in} Without calibration, the SA size must be $S(fr, \Delta V_{min})$. With N-fold redundancy, the SA size can be relaxed to $S(\sqrt[N]{fr}, \Delta V_{min})$. The relative individual SA size

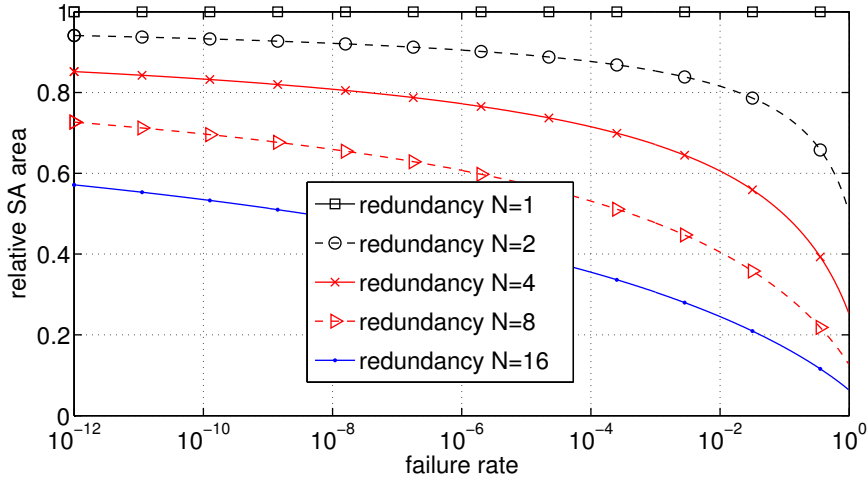


Figure 5.5: $N \cdot RS_N$, the sum of the sizes of the individual SAs when N-fold SA redundancy is used. This is a good measure for the area benefit provided by SA redundancy.

RS_N that is obtained by employing N-fold SA redundancy is then

$$\begin{aligned} RS_N(fr) &= \frac{S(\sqrt[N]{fr}, \Delta V_{min})}{S(fr, \Delta V_{min})} \\ &= \left(\frac{n(\sqrt[N]{fr})}{n(fr)}\right)^2 \end{aligned}$$

RS_N is independent of ΔV_{min}. Figure 5.4 shows RS_N, which is also representative for the SA energy consumption.

N SAs of this size are used, so $N \cdot RS_N(fr)$ is a good measure of the area reduction that is obtained through redundancy for a fixed allowed failure rate if overhead can be ignored. This ratio is shown in figure 5.5.

In [Ver08b], the improvement in failure rate for a fixed area was used as quality metric. As SA area is less critical in our design, it is of less importance to us.

5.4 SA tuning

5.4.1 Concept

Our design uses a novel technique: SA tuning. Here, each SA individually receives the most appropriate reference voltage based on its offset. Our design target is not better yield for a given area, but lower energy consumption for a fixed failure rate of 10^{-9}. Area was considered a lesser constraint, mainly because only one set of SAs is needed for the entire memory because of the use of extended global bit lines (see chapter 6).

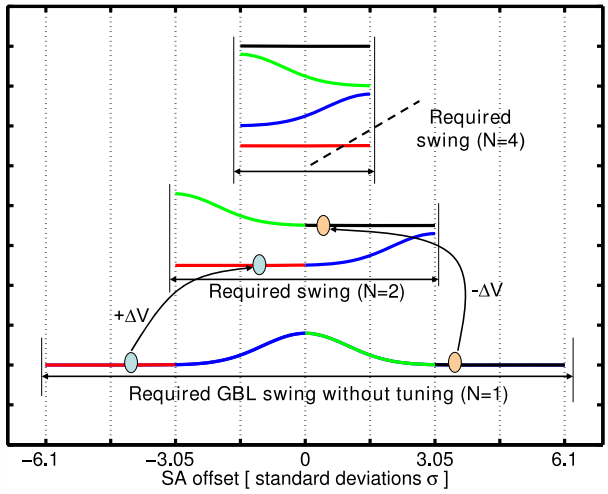


Figure 5.6: Effective offset of a tuned sense amplifier.

Fig. 5.6 illustrates how such a scheme changes the effective offset distribution for the SA after tuning. The probability density function (PDF) at the bottom is the offset distribution for a SA before calibration has been applied. If we accept a failure rate of 10^{-9}, we must only consider SAs whose native offset lies in the -6.1 to $+6.1$ σ range. Consider a simplified setup: a SA with one input connected to a single-ended BL and the other input connected to 0V. Without calibration, the BL voltage must be at least $+6.1\sigma_{V_{offset}}$ to ensure it is detected as a high input for all considered SAs. Conversely, the BL voltage must be below -6.1σ to ensure it is detected as a low input. If we assume the BL is precharged high and is pulled down conditionally, the minimal required single-ended BL swing is 12.2σ.

Now we introduce tuning with two voltage levels (N=2). If a fabricated SA has a negative native offset, the offset is shifted by $+\Delta V$. If the fabricated SA has a positive native offset, the offset is shifted by $-\Delta V$, with $\Delta V = 3.05\sigma$. The distribution after tuning with $N = 2$ is the sum of the two partial PDFs displayed in the figure. The effective offset range after tuning is only half of the native offset range. In our simplified setup, a single-ended BL swing of 6.1σ is sufficient. The most straightforward way to shift the offset voltage in the simplified setup is to change the voltage that is applied at the second input from $0V$ to + or - ΔV.

The same approach can be used with larger values of N. For a fixed SA design and sizing, the minimal required differential input signal after tuning is N times

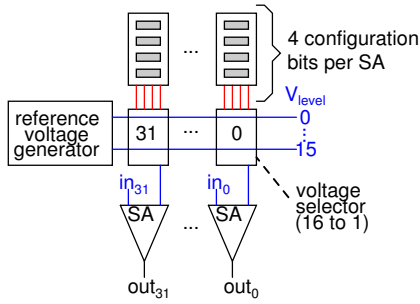


Figure 5.7: Setup for SA tuning (N=16 ; word length=32).

smaller than without tuning:

$$\Delta V_{min,\text{tuned}} = \frac{\Delta V_{min,\text{same SA, uncalibrated}}}{N}$$

Hence, tuning allows to sense a differential input signal V_{in} with a sense amplifier with native $DeltaV_{min,uncalibrated} = N \cdot V_{in}$. The SA size that is required to sense V_{in} with and without tuning is

$$\begin{aligned} S_{tuned} &= \left(n(fr) \frac{\sigma_{\Delta V_{T,min}}}{\Delta V_{in} \cdot N} \right)^2 \\ S_{traditional} &= \left(n(fr) \frac{\sigma_{\Delta V_{T,min}}}{\Delta V_{in}} \right)^2 \end{aligned}$$

The ratio between the two SAs sizes is

$$\frac{S_{traditional}}{S_{tuned}} = N^2 \tag{5.9}$$

This ratio is a good indication of the reduction in active energy that can be obtained with SA tuning compared to a traditional SA design.

Figure 5.7 illustrates the setup for tunable SAs with 4 configuration bits ($N = 16$). An on-chip reference voltage generator generates 16 reference voltages $V_{level,0} \ldots V_{level,15}$. Each SA is accompanied by 4 memory elements. The content of these memory elements controls a voltage selector which selects one of the 16 available voltage levels to be used as the reference voltage input for this SA. Tuning does not require any selection in the critical logic path, so it scales gracefully to large values of N.

5.4.2 Implementation

In this prototype implementation, the reference voltages are generated on-chip with a simple resistive divider as shown in fig. 5.8a, which consumes $13\mu A$ DC current. In our design, the high DC impedance of the voltage generator does

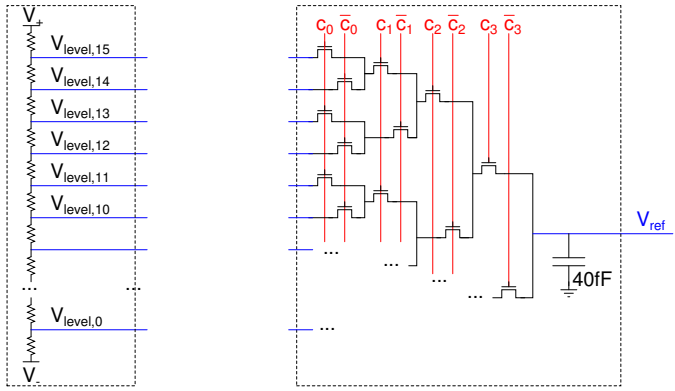


Figure 5.8: Prototype implementation of (a) reference voltage generator, and (b) voltage selector.

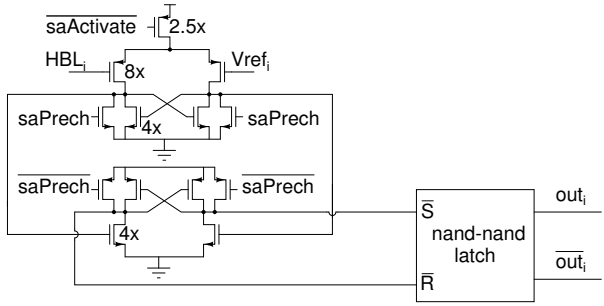


Figure 5.9: SA schematic. Transistor widths expressed in multiples of 120nm. All other transistors are 200nm wide. All transistors have minimal length ($L_{drawn} = 80nm$).

not pose a problem since the SAs that are used do not draw any DC current from their inputs. The voltage selectors are implemented as in fig. 5.8b. Notice the large decoupling capacitance at the output of each voltage selector. This reduces the AC-impedance of the reference voltage tree to an acceptable level to avoid excessive kick-back noise from one SA to the reference voltages of other SAs, which would result in data-dependent problems.

Fig. 5.9 is a schematic of the sense amplifier that is used in this design. Notice that the first stage draws static current as long as $\overline{saActivate}$ remains low. When the pulse width of $\overline{saActivate}$ is well controlled, this pulsed-mode SA consumes less energy than the self-stopping SA used in [Cos07].

The offset of the 32 SAs on a single die was measured. The distribution has a standard deviation of $\sigma_{V_{offset}} = 19mV$ and appears to be Gaussian, as expected. To obtain a failure rate of less than 10^{-9} with these SAs without the use of

Measured $\sigma_{V_{offset}}$	$19mV$
minimal tuning range $12.2\sigma_{V_{offset}}$	$232mV$
tuning range used	$300mV$
config bits used	4 ($N = 16$)
tuning step	$\approx 20mV$
minimal single-ended input swing measured without errors	$40mV$
total area overhead	$< 1\%$

Table 5.2: SA tuning: parameters and specs.

tuning would require a single-ended input swing of at least $2 \cdot 6.1 \cdot \sigma_{V_{offset}} = 230mV$. The novel design with tunable SAs employs a sufficient tuning range of $300mV$ and 16 levels, which in theory allows for a very small single-ended input swing of just under $20mV$. However, some margin is needed to guard against other noise sources and to obtain high speed operation. With relaxed timing settings ($2ns$ memory access), no errors were observed when reading with $40mV$ single-ended GBL swing ($\Delta V_{min} = 20mV$). In normal operation, $100mV$ swing is used and the SA reference voltages are chosen as to detect inputs below about $60mV$ as low logic inputs and inputs above $80mV$ as high inputs. This setting results in a faster access than the symmetric alternative, where the average threshold voltage would be at $50mV$.

The area of the tuning circuitry per SA is rather large. However, as the employed GBL architecture requires only 32 SAs, the total area overhead is less than 1%. Table 5.2 summarizes the main properties of the tuning system used.

5.4.3 Calibrating the tunable SAs

In the calibration phase, the best setting for the four calibration bits of each SA must be determined. In our system, the GBLs are precharged to $100mV$. If a cell stores a 1, the associated GBL will remain floating at $100mV$. If the cell stores a 0, the GBL will be pulled down by the readbuffer. As the driver transistor operates in the linear region, the resulting waveform is a decaying exponential that settles only slowly towards $0mV$. In a high performance memory, it is obviously not desirable to have to wait until the signal settles down. Rather, all signals sufficiently below $100mV$ should be sensed as low values. The optimal threshold depends on the noise on the floating GBLs and the required level of robustness. In the calibration method that is proposed in the next paragraphs, all GBL values above $80mV$ are treated as high values, everything below $60mV$ as low. In this text, the GBL voltage at which the outcome of SA sensing changes from high to low is called the SA's GBL threshold voltage.

During calibration, all GBLs are precharged to $100mV$, as in the normal operation mode. Fig. 5.10 shows the circuit setup. The reference voltage setting for all SAs is set to 0. All SAs are activated. As a wide enough tuning range

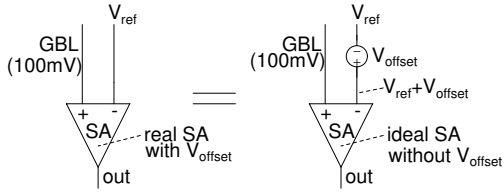


Figure 5.10: Setup during the calibration phase.

Table 5.3: Application of the calibration algorithm.

setting	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	lowest			GBL
V_{ref}[mV]	-80	-60	-40	-20	0	20	40	60	80	100	120	140	160	180	200	220	setting			toggle
native SA offset	sensing outcome with GBL @ 100mV																that results in '0'	setting to use	Vref used [mV]	voltage with tuning
19mV	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	9	7	60	79mV
1mV	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	9	7	60	61mV
-95mV	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	14	12	160	65mV
95mV	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	5	3	-20	75mV
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15				

is provided, all SAs should return a high SA output. When a very low V_{ref} is applied, $100mV$ is considered high, even when a large native offset is present. This step is repeated for each reference voltage setting up to 15. Table 5.3 shows the result of this sweep for four SAs with different native offset.

The lowest reference voltage setting for which the ith SA detects the $100mV$ on the GBL as a low value, is now called n_i. When the ith SA is used with voltage reference setting n_i, its GBL threshold voltage will always be somewhere between $100mV$ and $120mV$. When voltage reference setting $n_i - 2$ is applied to this i^{th} SA, the GBL threshold voltage is located between $60mV$ and $80mV$, as was desired. In the prototype implementation, this calibration loop was controlled off-chip. The calibration bits were set by means of an input shift register.

This algorithm could be implemented as a small FSM. It would require a 4-bit counter to loop over the 16 reference settings. The counter values are copied into the 4-bit calibration registers of those SAs whose output is still 1. After the loop is completed, all calibration bits contain the lowest setting that results in the detection of 0. We still need to subtract 2 from these values, which requires 32 4-bit subtract 2 units if full parallelism is desired. If so, calibration could be completed in slightly more than 16 clock cycles. In many systems, the calibration could also be delegated to software.

The method described here deviates slightly from what was discussed in [Cos08]. The main advantage of this new method is that it only requires the normal GBL precharge voltage and the reference voltages which are already needed during normal operation.

5.5 Asymmetric sense amplifier redundancy

In the SA redundancy scheme that was discussed up to now, all SAs were made identical to minimize area for a given failure rate. However, when energy is more important than area, it is beneficial to use different sizes for the SAs. For example, assume a setup with one small SA SA_1 and one big SA SA_2. If the small SA is designed with an allowed failure rate of $fr_1 = 5\%$, its $n_{fr_1} \approx 2$, and its energy consumption is $(\frac{6}{2})^2 = 9$ times smaller than that of a SA that is designed for a failure rate of $2 \cdot 10^{-9}$. If the redundant SA is designed for this failure rate, the combination will work for sure. On average (over many different instances), the energy consumption will be

$$E_{avg} = p_1 \cdot E_1 + p_2 \cdot E_2 = 95\% \frac{E_{big}}{9} + 5\% E_{big} = 0.16 E_{big}$$

Even with these random choices, the average energy consumption is lower than for symmetrical SA redundancy with $N = 4$.

We will now provide a slightly more complete description of asymmetric SA redundancy. N redundant SAs are used, with failure rates $fr_1 \ldots fr_N$. The indices are sorted on SA size so that the smallest SA has index 1. The failure rate after applying redundancy, fr, is

$$fr = P[\text{all N SAs fail}] = \prod_{i=1}^{N} fr_i$$

For our example with $N = 2$, $fr = 2 \cdot 10^{-9}$ and $fr_1 = 5\%$, we should use $fr_2 = \frac{fr}{fr_1} = 4 \cdot 10^{-8}$. This means that $n_2 = 5.5$ instead of $n_2 = 6.1$.

The probability that SA i of a redundant set is used, is called p_i. If the smallest SA works, it will be used:

$$p_1 = P[SA_1 \text{ works}] = 1 - P[SA_1 \text{ fails}] = 1 - fr_1 \tag{5.10}$$

The second SA is used only if it works correctly and the first SA does not work. These events are assumed to be independent, so

$$\begin{aligned} p_2 &= P[SA_1 \text{ fails } \& \ SA_2 \text{ works}] \\ &= P[SA_1 \text{ fails}] \cdot P[SA_2 \text{ works}] \\ &= fr_1 \cdot (1 - fr_2) \end{aligned}$$

In general, the ith SA is used if all small SAs fail and the ith SA itself works correctly.

$$p_i = (\prod_{j=1}^{i-1} fr_j) \cdot (1 - fr_i) \tag{5.11}$$

These probabilities do not add up to 1, as there is still a probability fr that there is no working SA in the redundant set. In the following graphs, this

remaining failure probability fr is added to p_N, the probability that the largest SA is used.

$$p_N = 1 - (\sum_{j=1}^{N-1} fr_j) \tag{5.12}$$

Figure 5.11 shows the results of a numerical optimisation for $N = 2$. For each fr, a sweep over all possible values for fr_1 is performed, the corresponding fr_2 is obtained from equation 5.12. The value of fr_1 that minimizes E_{avg} was selected. The area consumption is slightly smaller than for a traditional SA if the calibration overhead can be ignored. The energy consumption is significantly reduced. At $fr = 10^{-9}$, the relative energy consumption is 0.14, comparable to symmetrical redundancy with $N = 5$ (ignoring overhead).

The same exercise was performed for $N = 3$. In this case, a sweep is performed over fr_1 and fr_2, and fr_3 is obtained from equation 5.12. The results are shown in figure 5.12. At $fr = 10^{-9}$, the relative energy consumption is 0.061, comparable to symmetrical redundancy with $N = 10$ (ignoring overhead).

The same exercise was performed for $N = 4$. In this case, a sweep is performed over fr_1, fr_2 and fr_3, and fr_4 is obtained from equation 5.12. The results are shown in figure 5.13. At $fr = 10^{-9}$, the relative energy consumption is 0.035, comparable to symmetrical redundancy with $N = 15$ (ignoring overhead).

5.6 Summary

This chapter discussed SA calibration, a family of very effective techniques to overcome the traditional limitations on SA performance. Section 5.2 recapitulates these limitations and illustrates why it is important to overcome these limitations, especially for memories with a large word length and for memories in DDSM technologies.

Three calibration techniques are discussed. Section 5.3 reviews SA redundancy. section 5.4 introduces SA tuning, a novel type of SA calibration that we introduced in [Cos08, Cos09]. Section 5.5 introduces asymmetric SA redundancy, an interesting novel variation on SA redundancy that was not previously disclosed.

Table 5.4 summarizes the performance of these different techniques. Numerical values are provided for an allowed failure rate of 10^-9. The table provides two metrics, which are normalized against the values for a traditional SA without calibration. ΔV_{min} is the differential bit line voltage swing that can be accommodated by a fixed SA, while E_{SA} is the energy consumption of a SA (with fixed topology) that must accommodate a fixed differential bit line voltage swing. As discussed in section 5.2, Pelgrom's law imposes a simple relationship between both metrics:

$$E_{SA} \sim \left(\frac{1}{\Delta V_{min}}\right)^2$$

Example absolute values for traditional SAs can be found in chapter 4 and in table 5.1.

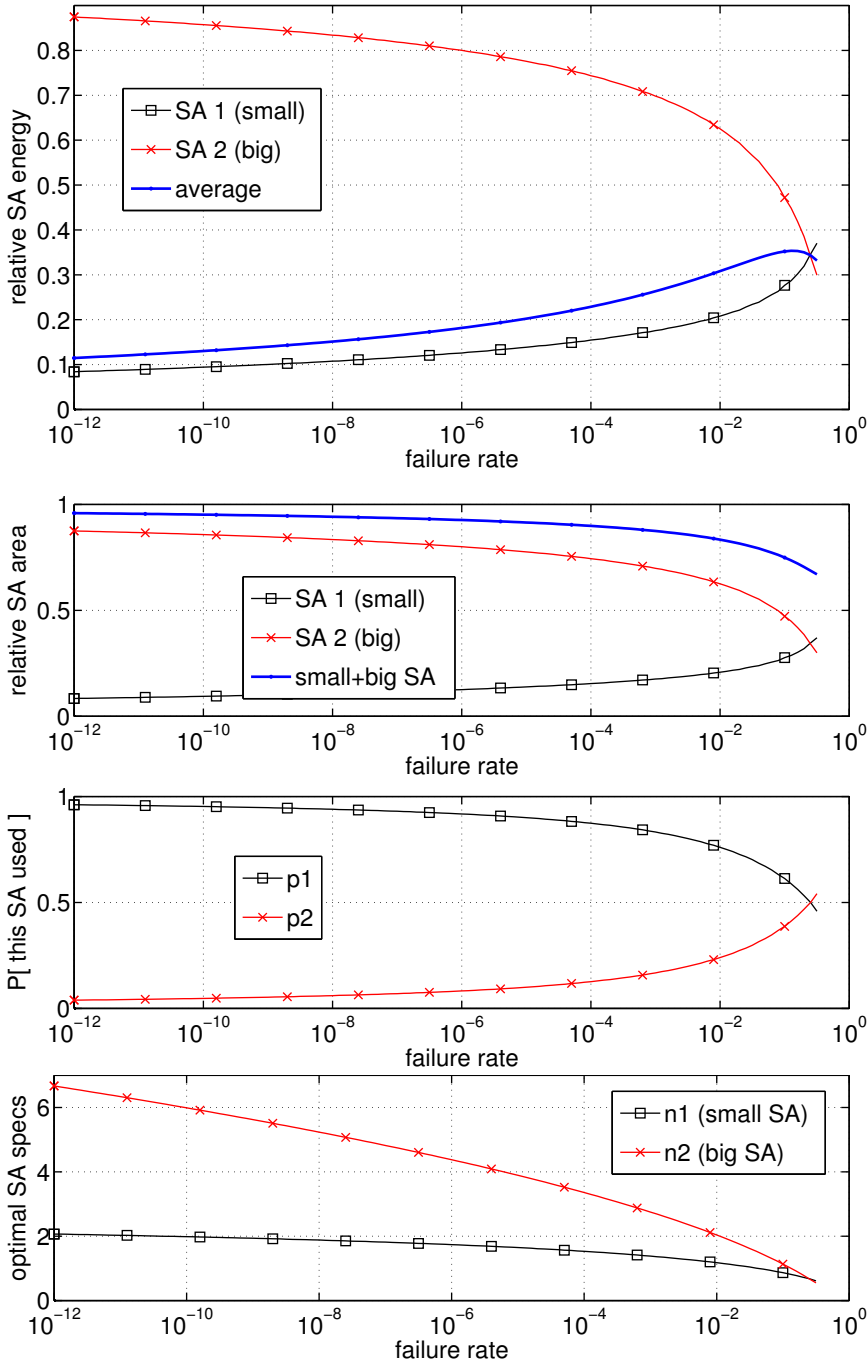


Figure 5.11: Optimal results for asymmetric SA redundancy with $N = 2$. Energy and area are normalised to those of a traditional SA with identical input signal and fr.

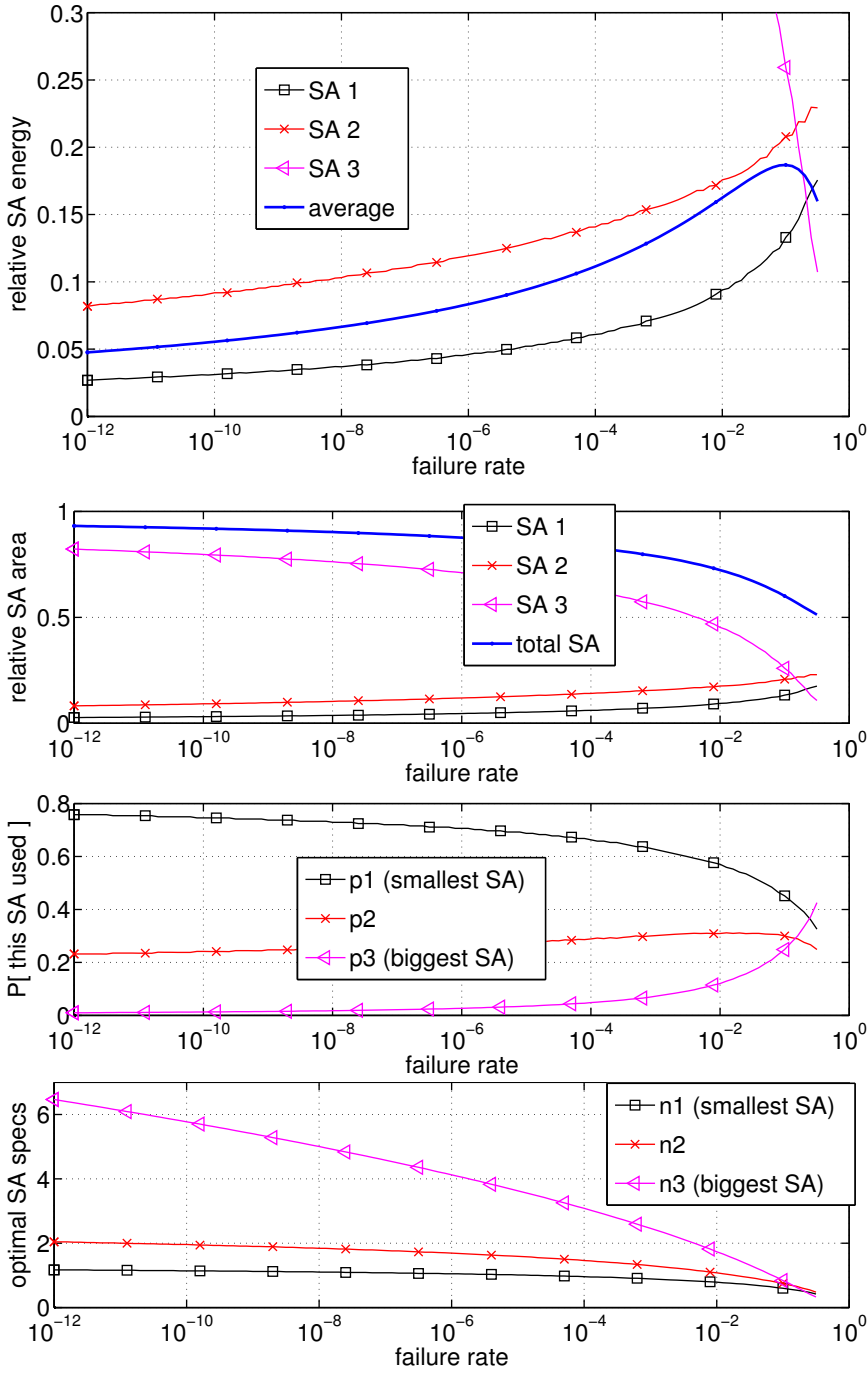


Figure 5.12: Optimal results for asymmetric SA redundancy with $N = 3$. Energy and area are normalised to those of a traditional SA with identical input signal and fr.

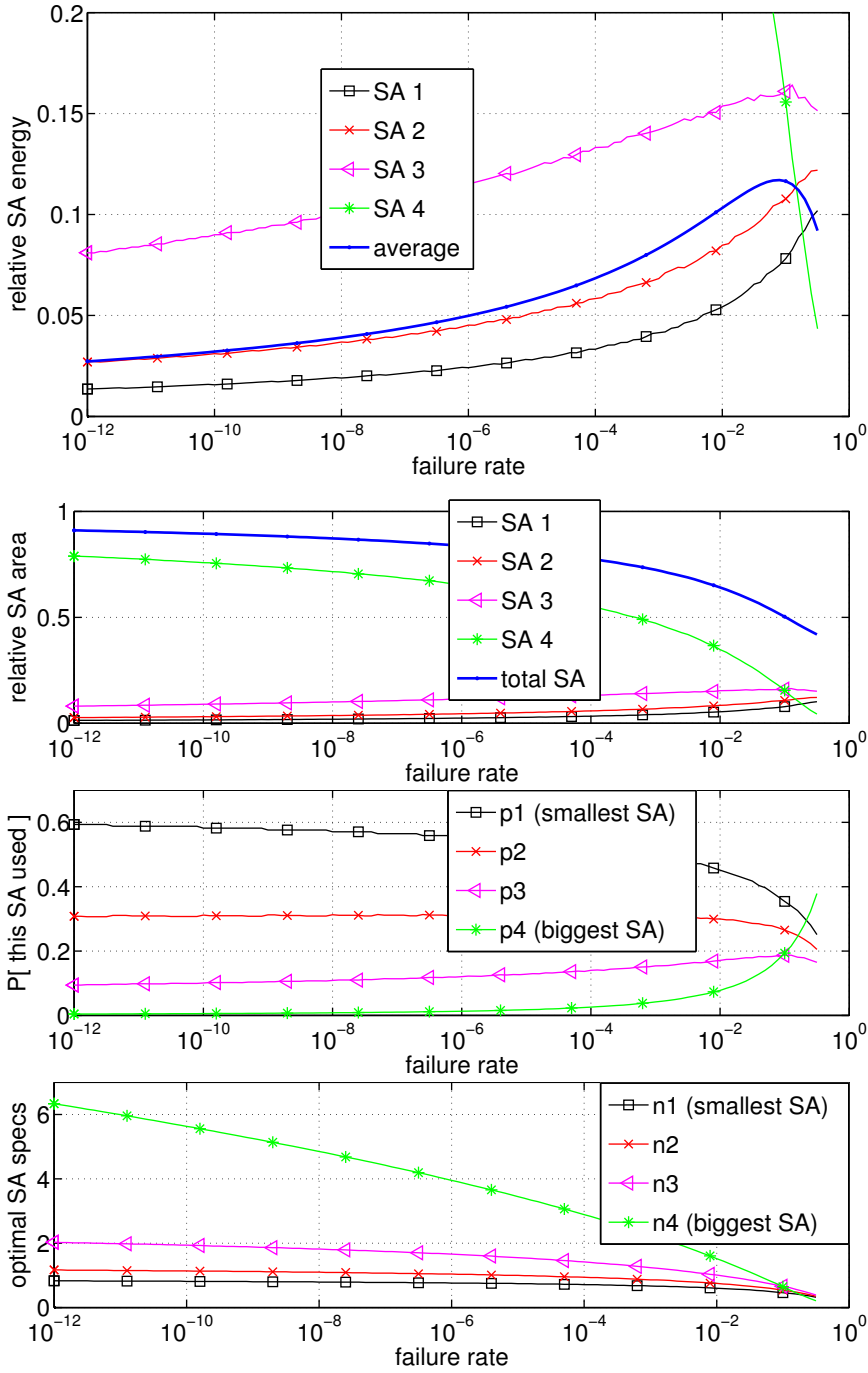


Figure 5.13: Optimal results for asymmetric SA redundancy with $N = 4$. Energy and area are normalised to those of a traditional SA with identical input signal and fr.

	required ΔV_{min}			required E_{SA}	
	$[\ \sigma_{V_{offset}}]$	$[\%]^a$	gain	$[\%]^b$	gain
without calibration	$n(fr)$	$\frac{n(fr)}{n(fr)}$		$\left(\frac{n(fr)}{n(fr)}\right)^2$	
	6.11	100.0%	1.0x	100.0%	1.0x
N-fold redundancy	$n(\sqrt[N]{fr})$	$\frac{n(\sqrt[N]{fr})}{n(fr)}$		$\left(\frac{n(\sqrt[N]{fr})}{n(fr)}\right)^2$	
$N=1$	6.11	100.0%	1.0x	100.0%	1.0x
$N=2$	4.16	68.1%	1.5x	46.4%	2.2x
$N=3$	3.29	53.9%	1.9x	29.0%	3.4x
$N=4$	2.77	45.3%	2.2x	20.5%	4.9x
$N=5$	2.41	39.5%	2.5x	15.6%	6.4x
$N=6$	2.15	35.2%	2.8x	12.4%	8.1x
$N=7$	1.94	31.8%	3.1x	10.1%	9.9x
$N=8$	1.78	29.1%	3.4x	8.5%	11.8x
$N=9$	1.64	26.9%	3.7x	7.2%	13.8x
$N=10$	1.53	25.1%	4.0x	6.3%	15.9x
$N=11$	1.43	23.4%	4.3x	5.5%	18.2x
$N=12$	1.35	22.1%	4.5x	4.9%	20.6x
$N=13$	1.27	20.8%	4.8x	4.3%	23.0x
$N=14$	1.21	19.7%	5.1x	3.9%	25.6x
$N=15$	1.15	18.8%	5.3x	3.5%	28.3x
$N=16$	1.09	17.9%	5.6x	3.2%	31.2x
N-fold tuning	$\frac{n(fr)}{N}$	$\frac{1}{N}$		$\left(\frac{1}{N}\right)^2$	
$N=1$	6.11	100.0%	1.0x	100.0%	1.0x
$N=2$	3.05	50.0%	2.0x	25.0%	4.0x
$N=3$	2.04	33.3%	3.0x	11.1%	9.0x
$N=4$	1.53	25.0%	4.0x	6.2%	16.0x
$N=5$	1.22	20.0%	5.0x	4.0%	25.0x
$N=6$	1.02	16.7%	6.0x	2.8%	36.0x
$N=7$	0.87	14.3%	7.0x	2.0%	49.0x
$N=8$	0.76	12.5%	8.0x	1.6%	64.0x
$N=16$	0.38	6.2%	16.0x	0.4%	256.0x
asymmetric redundancy	numerical optimisation			numerical optimisation	
$N=2$	2.28	37.4%	2.7x	14.0%	7.1x
$N=3$	1.50	24.6%	4.1x	6.1%	16.5x
$N=4$	1.15	18.8%	5.3x	3.5%	28.4x

Table 5.4: Overview table of the performance of the calibration techniques presented in this chapter. Numerical values are given for a failure rate $fr = 10^{-9}$.

[a] Compared to a SA without calibration with same E_{SA}
[b] Compared to a SA without calibration for same ΔV_{min}

The different techniques come with different implementation costs and limitations, which makes it impossible to indicate a clear winner. SA redundancy requires a selection in the critical path, which imposes a penalty in the dynamic energy consumption and delay of the SA which increases with larger values of N. On the other hand, the sensitivity of SA tuning to voltage fluctuations between the bit line precharge voltage and the reference voltage might be a source of concerns about robustness. Alternative implementations of SA tuning can probably mitigate these issues, but have not been studied in this work.

Two effects put fundamental limitations on the SA specifications ΔV_{min} and E_{SA} that can be obtained by calibration: the minimal transistor size of the technology puts a lower bound on E_{SA}, while thermal noise puts a lower bound on the combination of E_{SA} and ΔV_{min}. The impact of thermal noise on SAs was discussed in sections 4.2.3 and 4.6.7. For typical SA implementations as used in low power memories in a $90nm$ technology, thermal noise increases the minimal required ΔV_{min} by $5mV$ to $20mV$.

Contributions and highlights

This chapter discusses the growing need for SA calibration and compares three different calibration methods. The two most efficient calibration schemes, SA tuning and asymmetric SA redundancy are original contributions of this work. Asymmetric SA redundancy reduces the required bit line swing with an additional factor of two compared to symmetric SA redundancy. SA tuning introduces less energy and delay overhead and allows in principle for a larger reduction in bit line swing than asymmetric redundancy, but it has its own implementation difficulties. Both new calibration approaches enable the use of a 5 times smaller bit line swing for a given SA energy consumption.

Chapter 6

Data Transfers

6.1 Introduction

This chapter addresses the transfer of the data bits between the memory cell and the memory output interface during read and from the memory input interface to the memory cell during write.

Section 6.2 discusses the read operation. First, a comparison between complementary and single-ended sensing is provided. Second, a number of practical implementations are presented and their strong and weak points are highlighted. Third, two implementations are analysed and compared quantitatively including the effects of intra-die variations. Fourth, different precharge control schemes for the bit lines are outlined.

Section 6.3 addresses the write operation. The traditional setup is outlined. The write energy in the traditional scheme is problematically large, so several techniques to reduce the write energy are discussed.

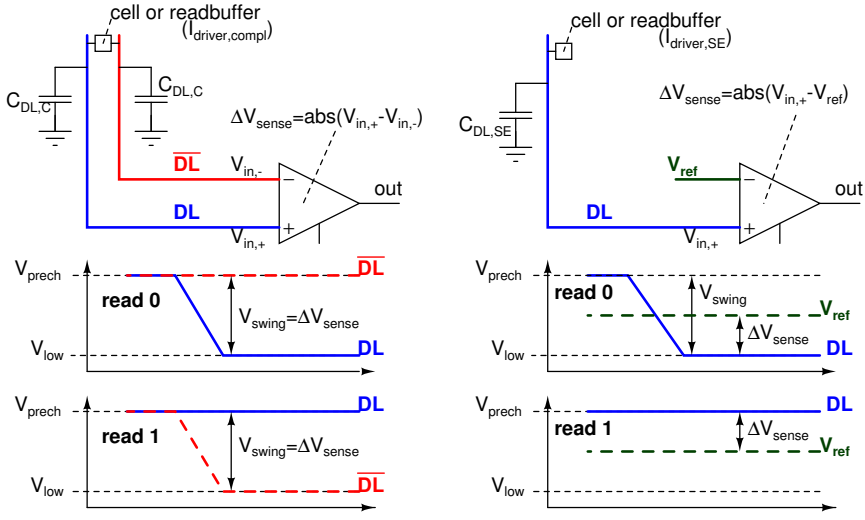
Section 6.4 summarises the most important findings of this chapter.

6.2 Read

6.2.1 Signalling types

Almost all published memories use one of the signalling schemes from figure 6.1 for their read bit lines. In this figure, I_{driver} can represent either the cell read current I_{read} or the current from a local amplification circuit such as the readbuffer. The data lines DL represent the sum of all the different bit lines that connect the cell or readbuffer to the global read sense amplifiers. When complementary signalling is used as in figure 6.1a, both data lines DL and $\overline{DL}$ of the data line pair are precharged to V_{prech}. Based on the cell state, one bit line of the bit line pair is pulled down by an amount V_{swing}, from V_{prech} to V_{low}. V_{low} is the voltage headroom that is available for the driver. The larger V_{low}, the easier it is to obtain a large driver current I_{driver}. V_{swing} must be larger than ΔV_{min}, the smallest input voltage difference that the SA can sense reliably. Assuming a constant current $I_{driver,compl}$ for the cell or readbuffer, the delay is

$$D_{compl} = \frac{Q_{DL,C}}{I_{driver,compl}} = \frac{C_{DL,C} \cdot \Delta V_{min}}{I_{driver,compl}}$$



(a) Complementary signalling.

(b) Single-ended signalling.

Figure 6.1: Signalling methods.

When all charge is taken from the normal power supply Vdd, the energy consumption is

$$E_{compl,Vdd} = C_{DL,C} \cdot \Delta V_{min} \cdot Vdd$$

If a reduced voltage Vdd_{low} is available for the precharge voltage,

$$E_{compl,Vdd_{low}} = C_{DL,C} \cdot \Delta V_{min} \cdot Vdd_{low}$$

The lowest voltage that can be used for Vdd_{low} is $V_{prech} = V_{low} + \Delta V_{min}$, so

$$E_{compl,Vdd_{low}} \geq C_{DL,C} \cdot \Delta V_{min} \cdot (V_{low} + \Delta V_{min})$$

When single-ended signalling is used, the signal is transferred by a single data line DL. This bit line is precharged to V_{prech}. If the state of the accessed cell is "0", DL is pulled down by an amount V_{swing}, from V_{prech} to V_{low}. If the state is "1", the bit line stays at V_{prech}. The SA uses a reference voltage V_{ref}. The margin for sensing is maximised when V_{ref} is halfway between V_{prech} and V_{low}. The input voltage difference that is available for the SA is then $\frac{V_{swing}}{2}$, hence V_{swing} must be larger than $2 \cdot \Delta V_{min}$, with ΔV_{min} the smallest input voltage difference that the SA can sense reliably. Assuming a constant current $I_{driver,SE}$ for the cell or readbuffer, the delay is

$$D_{SE} = \frac{Q_{DL,SE}}{I_{driver,SE}} = \frac{C_{DL,SE} \cdot 2 \cdot \Delta V_{min}}{I_{driver,SE}}$$

The average energy consumption of single-ended sensing schemes depends linearly on p_0, the probability that a cell contains a "0". When all charge is taken

from the normal power supply Vdd, the average energy consumption is

$$E_{SE,Vdd} = p_0 \cdot C_{DL,SE} \cdot 2 \cdot \Delta V_{min} \cdot Vdd$$

If a reduced voltage Vdd_{low} is available for the precharge voltage,

$$E_{SE,Vdd_{low}} = p_0 \cdot C_{DL,SE} \cdot 2 \cdot \Delta V_{min} \cdot Vdd_{low}$$

The lowest voltage that can be used for Vdd_{low} is $V_{prech} = V_{low} + 2 \cdot \Delta V_{min}$, so

$$E_{SE,Vdd_{low}} \geq p_0 \cdot C_{DL,SE} \cdot 2 \cdot \Delta V_{min} \cdot (V_{low} + 2 \cdot \Delta V_{min})$$

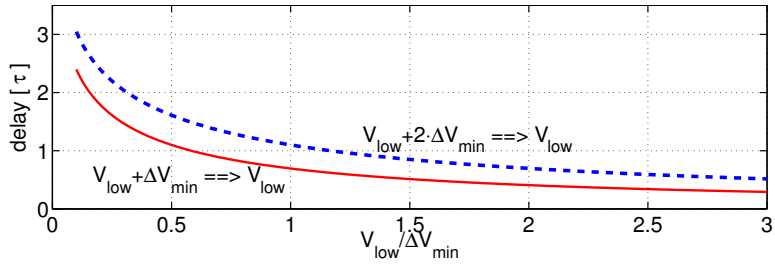
It is a rather subtle exercise to assess the difference in energy and delay between the two signalling schemes because all parameters differ to some extent. The next paragraphs discuss the differences in C_{DL} and I_{driver} between the two schemes.

For single-ended signalling, only one signal wire must be routed per cell pitch. If a significant cross-coupling capacitance exists between the signal wires, shield wires must be inserted to ensure signal integrity. When the thin cell layout style is used, shield wires are probably not necessary as the wire pitch is large ($19F$, see section 3.2) and other capacitances to wires at static voltages will dominate.

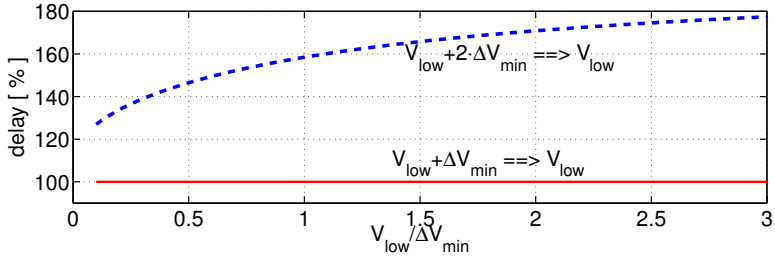
For complementary signalling, two signal wires must be routed per cell pitch. Without countermeasures, the cross-coupling capacitance will likely dominate. This can be addressed by inserting shield wires or by using bit line folding. With shield wires, 4 wires must be routed per cell pitch. Unless more than one metal layer is available to route the signal wires, this will increase the capacitance significantly. $C_{DL,compl} \approx 2 \cdot C_{DL,SE}$ is a reasonable assumption in this case. With correct bit line folding (also called bit line twisting) [Yos88], the voltage drop on all signal wires that must remain high can be made independent of the data pattern, and certainly smaller than $\frac{V_{swing}}{2}$. A reasonable estimate of the voltage drop is $\frac{V_{swing}}{4}$. The available ΔV_{sense} is reduced by this voltage drop. This must be compensated by a larger V_{swing}.

In case the driver is a readbuffer, its size can be twice as wide in a single-ended signalling scheme than in a complementary signalling scheme without increase in leakage, area and dynamic energy (if $p_0 \leq 50\%$). In this case, a fair comparison assumes $I_{driver,SE} \approx 2 \cdot I_{driver,compl}$. When the driver is a 6T memory cell, the situation is different. The cell area and leakage normally dont decrease significantly when a single-ended read scheme is adopted. In this case, $I_{driver,SE} \approx I_{driver,compl}$.

Figure 6.2 highlights another point that should be considered when comparing the delay of single-ended and complementary signalling. The fact that the required V_{swing} is twice as large for single-ended as for complementary signalling increases the delay. If the driver transistor operates in the saturation region, the delay will be twice as large. However, when the driver transistor operates in the linear region with its source at $0V$, the voltage waveform on the signal



(a)



(b)

Figure 6.2: Delay that is required for the discharge from $V_{low}+2\cdot\Delta V_{min}$ to V_{low} and for the discharge from $V_{low}+\Delta V_{min}$ to V_{low}. When V_{low} is small compared to ΔV_{min}, the discharge from $V_{low}+2\cdot\Delta V_{min}$ to V_{low} takes significantly less time than the discharge from $V_{low}+\Delta V_{min}$ to V_{low}.

wire evolves towards $0V$ as $V_{prech} \cdot exp(\frac{-t}{\tau})$, with $\tau = R_{driver} \cdot C_{DL}$. When V_{low} is small, the initial discharge from $V_{low} + 2 \cdot \Delta V_{min}$ to $V_{low} + \Delta V_{min}$ is much faster than the discharge from $V_{low} + \Delta V_{min}$ to V_{low}. Hence, the increase in delay is much smaller than a factor of 2. This situation is relevant to us, as the driver transistor in the readbuffer in section 6.2.2.3 typically operates in the linear region and V_{low} is chosen as low as the delay constraint allows.

If p_0 is expected to be around 50%, the two schemes are rather comparable, both in terms of delay and energy, and the optimal trade-off will depend on low-level implementation details. When p_0 is expected to be significantly below 50% as in [Cha05b], the advantages of single-ended signalling become very pronounced.

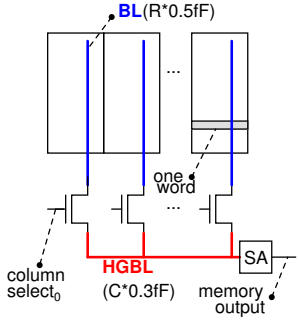
6.2.2 Overview of solutions

Figure 6.3 gives a high-level overview of different ways in which the data transfer from cell to memory output can be organised. Fully subdivided word lines are assumed. $HGBL$ is the horizontal global bit line, $VGBL$ is the vertical global bit line. For a $90nm$ technology and a thin cell layout, $C_{LBL,cell} = 0.5fF$, $C_{VGBL,cell} = 0.15fF$ and $C_{HGBL,cell} = 0.3fF$ are reasonable capacitance estimates. The capacitance overhead that is introduced by the bit line periphery still needs to be added to these numbers. In this section, T is the number of cells per local bit line, R is the number of rows in the memory and C is the number of cell columns. All solutions can be implemented both with complementary bit lines and with single-ended bit lines. In this section, only implementations with complementary bit lines are considered. As this text focuses on low-power memories, domino-evaluate read solutions as discussed in section 3.5.2 are not shown.

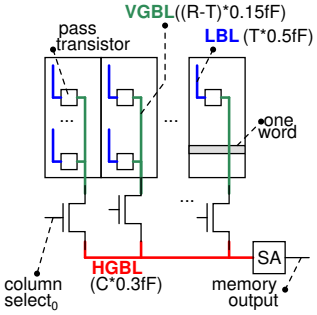
Figure 6.4 shows that the layout of the connectors between the vertical and horizontal bit lines requires some care. When the connectors are placed as in figure 6.4a, the effective length of $HGBL_0$ is increased by $L_{extra} \cdot M$, with M the number of word columns. L_{extra} is the height of the horizontal bit line bus. If the effective $HGBL$ pitch is half of that of $VGBL$, $L_{extra} = W_{column}/2$ and C_{HGBL} increases with 50%. When the connectors are placed as in figure 6.4b, C_{VGBL} increases, but this increase is not proportional to M. Unless M is very small, this second approach is preferable.

6.2.2.1 Solution A

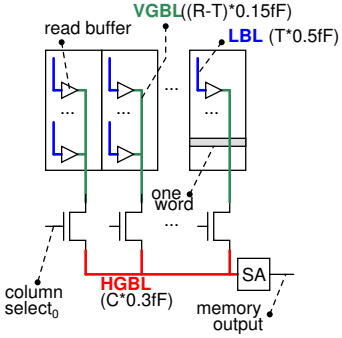
In solution A (figure 6.3a), all cells of a column connect to a single bit line pair. The bit line pair connects to the horizontal global bit lines through pass transistors. During a read access, BL and $HGBL$ are precharged high. When the word line is activated, the cell current I_{read} discharges BL and $HGBL$. This generates a voltage difference at the input of the SA, which amplifies this input towards the memory output. The precharge voltage must be above $Vdd_{WL} - V_{Tn,access}$ as the cell is otherwise destabilised (see figure 2.53). The pass transistors between BL and $HGBL$ must hence contain a PMOS transistor. For the write operation, BL must be pulled to ground. This requires NMOS



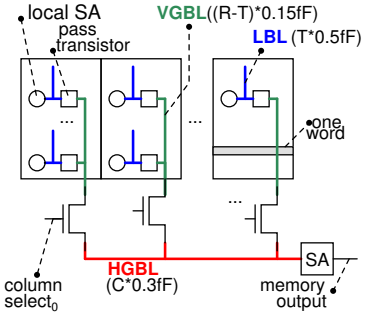
(a) solution A (no $VGBL$)



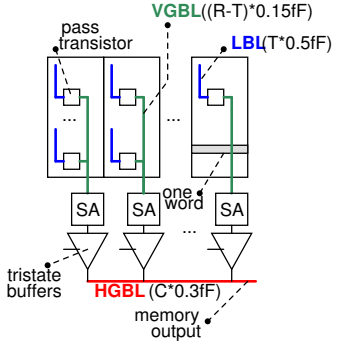
(b) solution B (pass transistor)



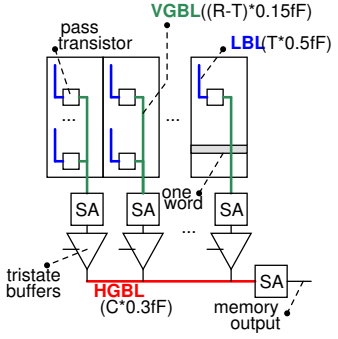
(c) solution C (readbuffer)



(d) solution D (local SA)



(e) solution E (per-column SA)



(f) solution F (per-column SA and low-swing $HGBL$)

Figure 6.3: Solutions to transfer data from the cell to the memory output using hierarchical bit lines and fully subdivided word lines. Capacitance estimates are for a $90nm$ technology.

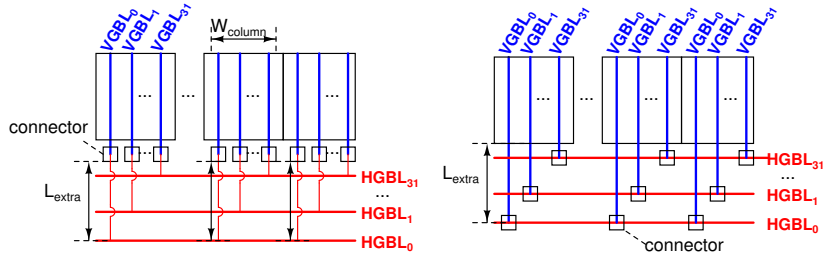


Figure 6.4: Two possible position schemes for the connectors between the vertical and horizontal bit lines.

transistors, so complementary pass transistors are required. In this solution, a reduced $V_{BL,precharge}$ increases the resistance of the pass transistors. In this section, $V_{BL,precharge}$ will be fixed at Vdd for all solutions that include a pass transistor connection to the local bit line.

In this solution, all bit line charge is sinked to $VssCell$ by the access and pull-down transistors of the cell. The energy consumption is hence

$$\begin{aligned} E_{BL,A} &= (Q_{BL} + Q_{HGBL}) \cdot Vdd + E_{control} \\ &\quad (C_{BL} \cdot V_{swing,BL} + C_{HGBL} \cdot V_{swing,HGBL}) \cdot Vdd + E_{control} \end{aligned}$$

Here, $C_{BL} \approx R \cdot C_{LBL,cell}$ and $C_{HGBL} \approx C \cdot C_{HGBL,cell}$. These capacitance values increase somewhat due to the bit line periphery. $E_{control}$ is the energy that is needed to control the gates of the pass transistors. The SA energy and the word line energy could be added to this value, but they are common to all solutions that are evaluated in this section, so they are not considered here. $V_{swing,HGBL}$ must be large enough to allow reliable sensing by the employed sense amplifier: $V_{swing,HGBL} \geq \Delta V_{min}$. As there is always some voltage drop over the pass transistor between BL and $HGBL$, $V_{swing,BL}$ is larger than $V_{swing,HGBL}$. If plenty of time is available, the word line can be disabled before the SA is enabled, and $V_{swing,BL} \approx V_{swing,HGBL}$. If the word line is enabled too long, both voltage swings will be significantly larger than ΔV_{min}, which increases the energy consumption.

6.2.2.2 Solution B

In solution B (figure 6.3b), T cells connect to a local bit line LBL. LBL is connected through a pass transistor to $VGBL$, which in turn connects through a pass transistor to $HGBL$. The main advantage of this scheme is the reduced total bit line capacitance: the vertical bit line capacitance can be reduced to about 50% of the value in solution A (figure 1.8). However, when C_{HGBL} is of the same magnitude as C_{VGBL}, the total GBL capacitance will only be reduced to about 75% of the value in solution A. The actual energy and delay reduction

will be smaller due to the voltage drop over the additional pass transistor and due to the additional control overhead. This text assumes $V_{BL,precharge} = Vdd$ for solution B as well. This solution has similar problems with swing control as solution A. The energy consumption of this solution is

$$\begin{aligned} E_{BL,B} &= (Q_{LBL} + Q_{VGBL} + Q_{HGBL}) \cdot Vdd + E_{control} \\ &= (C_{LBL} \cdot V_{swing,LBL} + C_{VGBL} \cdot V_{swing,VGBL} + \\ &\quad C_{HGBL} \cdot V_{swing,HGBL}) \cdot Vdd + E_{control} \end{aligned}$$

$C_{LBL} \approx T \cdot C_{LBL,cell}$, $C_{VGBL} \approx (R-T) \cdot C_{VGBL,cell}$ and $C_{HGBL} \approx C \cdot C_{HGBL,cell}$. These capacitance values increase further due to the additional bit line periphery. Especially the increase in C_{VGBL} due to the $\frac{R}{T}$ local bit line peripheral circuits that connect to each $VGBL$ is often significant. $E_{control}$ is the energy that is needed to control the gates of the pass transistors. This value is higher than in solution A, as twice as many pass transistors must be controlled.

6.2.2.3 Solution C

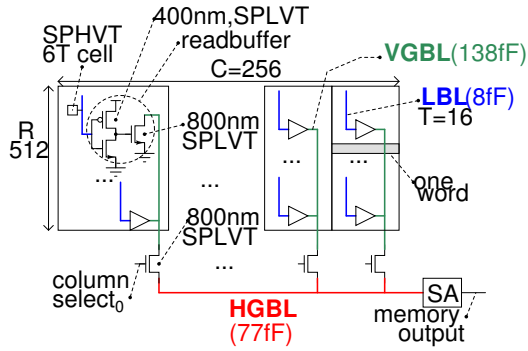
Solution C (figure 6.3c) is the short buffered local bit line with extended global bit lines scheme that was used in our designs [Cos07, Cos09]. The impact of this scheme on the memory cell is discussed in section 3.5.6. T memory cells connect to a local bit line, which is precharged high. When the word line is activated, the cell discharges the local bit line from $V_{LBL,precharge}$ to a value below $V_{trip,readbuffer}$, typically $0V$. When the voltage on LBL is below $V_{trip,readbuffer}$, the readbuffer starts to discharge $VGBL$, which is connected to $HGBL$ through a pass transistor.

In this setup, the precharge voltage of the global bit lines is not constrained by cell stability, so it can be precharged to a low voltage $V_{GBL,precharge}$, e.g. $200mV$. This has a number of advantages. First, it allows the use of NMOS pass transistors for read operations, which avoids the need for complementary pass transistors. Second, the voltage swing on the GBLs is now well controlled without having to rely on accurate timing control. Third, if a lower supply voltage Vdd_{low} is supplied separately, or generated with a DC-DC converter, a significant further reduction in bit line energy consumption can be obtained. If $Vdd = 1V$ and $Vdd_{low} = 200mV$, the GBL energy consumption reduces with an additional factor of 5. This has not been addressed in our work though.

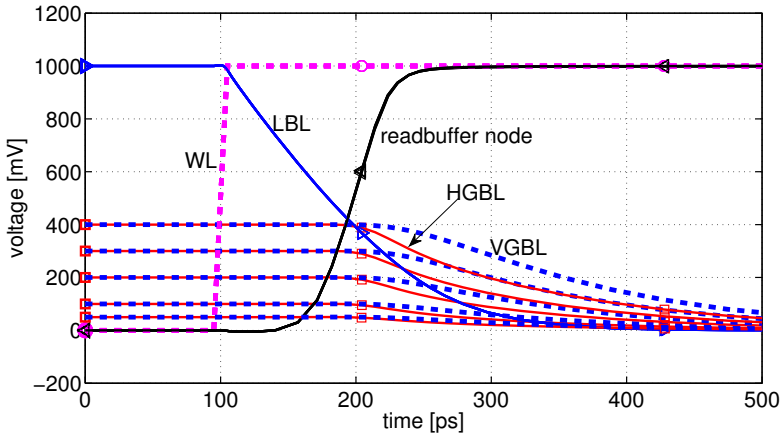
The energy consumption of solution C is

$$\begin{aligned} E_{BL,C} &= Q_{LBL} \cdot Vdd + (Q_{VGBL} + Q_{HGBL}) \cdot Vdd_{low} + E_{control} \\ &= C_{LBL} \cdot Vdd^2 + (C_{VGBL} \cdot V_{swing,VGBL} + \\ &\quad C_{HGBL} \cdot V_{swing,HGBL}) \cdot Vdd_{low} + E_{control} \end{aligned}$$

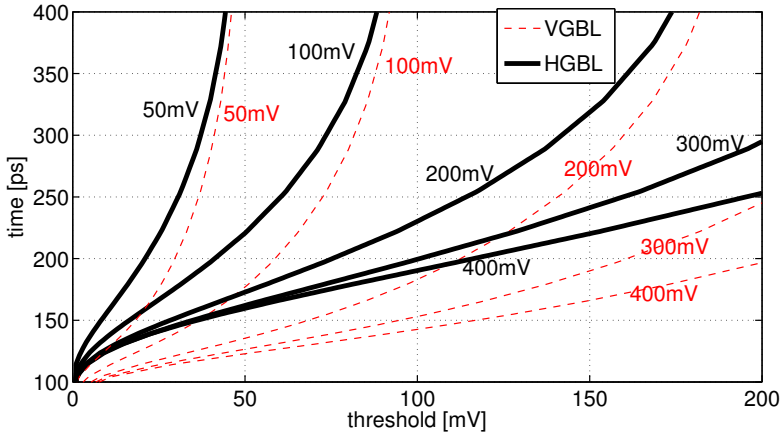
Unless a separate lower supply is provided, $Vdd_{low} = Vdd$. Because a full-swing signal is used on LBL, the LBL energy consumption is higher than in solution B. In $90nm$ with $T = 16$ at $Vdd = 1V$, the LBL energy is $8fJ$.



(a)



(b)



(c)

Figure 6.5: An example setup with readbuffer (solution C). All cell transistors are SPHVT transistors with $W = 200nm$. $R = 512$, $C = 256$, $T = 16$, $Vdd = 1V$. Capacitance estimates assume $C_{LBL,cell} = 0.5fF$, $C_{VGBL,cell} = 0.15fF$, $C_{HGBL,cell} = 0.3fF$. C_{VGBL} was increased with $\frac{R}{T} \cdot 2fF$ to account for the area increase and capacitance due to the local bit line periphery.

The optimal value of $V_{GBL,precharge}$ involves a trade-off between energy consumption and delay. Figure 6.5 provides an example. Figure 6.5c shows the delay between word line activation and the time at which the swing on the bit line matches the given threshold voltage. This delay includes the local bit line and read buffer delay. For example, assume that ΔV_{min} of the SA is $50mV$ and complementary bit lines are used. With $V_{GBL,precharge}$ below $50mV$, this swing is never obtained. The delay is $221ps$ with $V_{GBL,precharge}$ at $100mV$ and $172ps$ at $200mV$.

Figure 6.5b and 6.5c indicate that the access transistor between $VGBL$ and $HGBL$ has a significant negative impact on the read delay. This can be addressed by further increasing the size of this pass transistor, but unless its gate activity can be reduced by exploiting access locality, the control energy will increase significantly. Because the pass transistors operate at reduced V_{gs}, the negative impact of these pass transistors increases further when the supply voltage is reduced, unless transistors with a lower threshold voltage than that of the cell transistors can be used.

In solution C, the cell needs to sink a charge

$$Q_C = T \cdot C_{LBL,cell} \cdot (V_{LBL,precharge} - V_{trip,readbuffer}) = T \cdot C_{LBL,cell} \cdot \Delta V_{buffer}$$

while in solution A,

$$Q_A \approx (R \cdot C_{BL,cell} + C \cdot C_{HGBL,cell}) \cdot \Delta V_{min}$$

The impact of I_{read} on the total read delay is smaller in solution C than in solution A if $Q_C < Q_A$, hence

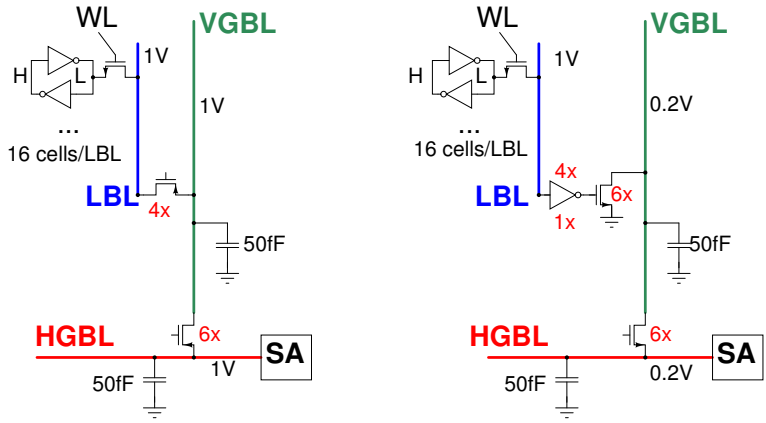
$$T < \frac{R \cdot C_{BL,cell} + C \cdot C_{HGBL,cell}}{C_{LBL,cell}} \cdot \frac{\Delta V_{min}}{\Delta V_{buffer}}$$

With $C_{BL,cell} = C_{LBL,cell} = 0.5fF$ and $C_{HGBL,cell} = 0.3fF$, this condition simplifies to $T < (R + 0.6 \cdot C) \cdot \frac{\Delta V_{min}}{\Delta V_{buffer}}$. For example, if $\Delta V_{min} = 50mV$, $\Delta V_{buffer} = 500mV$, $R = 256$ and $C = 256$, the impact of I_{read} is smaller in solution C than in A if T is smaller than 40.

A more complete comparison that includes the impact of transistor variations is provided in section 6.2.3.

6.2.2.4 Solution D

In solution D (figure 6.3d), a local sense amplifier (LSA) is added to the local bit line. Structurally, this local setup is similar to that in [Yan05] (see section 3.5.4), but in that design, the LSA is only used during write. Solution D also uses this LSA to speed up the read operation. The LSA must be triggered by an explicit strobe signal. The strobe signal must arrive after the slowest cell has developed a voltage swing of at least $\Delta V_{min,LSA}$ on the LBL, but if the strobe signal arrives too late, it does not speed up the delay. This more complex timing and the larger control energy are the main disadvantages of this approach. Typical



(a) Without readbuffer (b) With readbuffer

Figure 6.6: Simulation setup to compare extended GBL schemes with and without readbuffer. The capacitances in this test case are representative for a $90nm$ memory which is slightly smaller than 256x256 cells. Precharge voltages are displayed next to the wires. Cell supply voltage is $1.2V$, all other supplies are at $1V$. Transistor sizings are displayed next to the transistors in multiples of $120nm$. Both cells are traditional 6T cells with all SPHVT transistors with $width = 220nm$ and $length = 90nm$. All other transistors are SPLVT.

values for $\Delta V_{min,LSA}$ in $90nm$ are between $150mV$ and $200mV$. The main advantage of this approach is that it speeds up the memory access and reduces the impact of I_{read}. The LSA can also serve as low-swing write receiver (see section 6.3), which makes it a reasonably area-efficient solution.

6.2.2.5 Solution E and F

In solution E and F (figure 6.3e and 6.3f), the pass transistor between $VGBL$ and $HGBL$ is avoided by using a SA for each column. Solution E uses full-swing signals on $HGBL$ and is hence not a low power solution. Solution F has been used in several designs, e.g. [Yan05] and [See91]. Such schemes can be useful in larger memories or in situation where pass transistors between $VGBL$ and $HGBL$ would limit voltage scaling.

6.2.3 Analysis under intra-die variations

This section shows with an example that the insertion of a readbuffer between LBL and GBL reduces the impact of the cell read current I_{read} on the memory speed. This becomes especially beneficial in technology nodes with large intra-die variations. In these technology nodes, the cell read current varies widely because cell transistors must be kept as small as possible to preserve area. Another problem for I_{read} occurs with low-power designs: to limit the amount of standby leakage, the cells must be implemented using high threshold transistors.

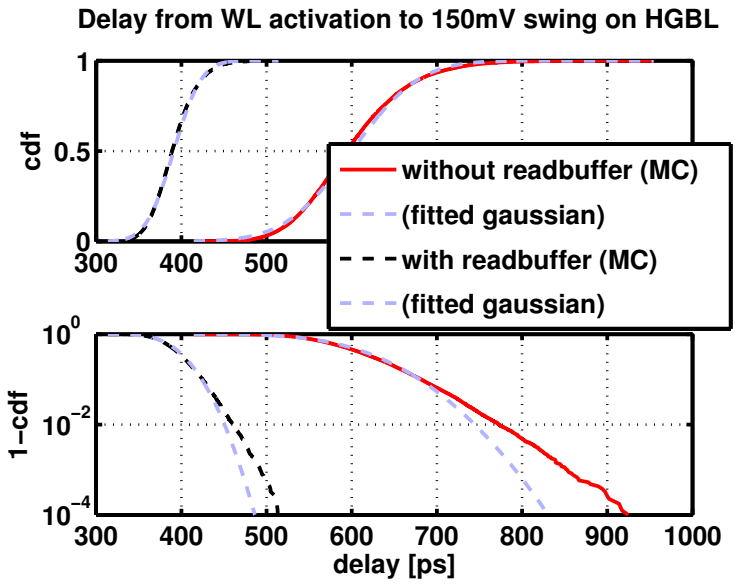


Figure 6.7: Distributions of the delay from WL activation to $150mV$ swing on $HGBL$ for the two setups in figure 6.6 as obtained by performing 20K Monte Carlo runs. Gaussian distributions have been fitted to both full datasets. All curves are displayed both on a linear and a logarithmic scale.

This reduces cell read current, and further aggravates the issue of variations. The readbuffer current suffers less from these intra-die variations for two reasons. The readbuffer does not have to use high threshold transistors, and because the size of the readbuffer transistors has less impact on the memory area, they can be made somewhat larger.

Figure 6.6 shows the two simulation setups for which the performance under intra-die variations will be compared in the next paragraphs. In the setup with readbuffer, GBL is precharged to $0.2V$. In both setups, it is assumed that the SAs require an input swing of $150mV$ to perform reliable sensing, so delay is measured from WL activation to $150mV$ swing on $HGBL$.

20000 Monte Carlo (MC) runs were performed to obtain a good impression of the impact of intra-die variations. Figure 6.7 shows the resulting cumulative distribution of the delay for both setups. The nominal delay for the setup without readbuffer is $600ps$, for the setup with readbuffer, it is only $400ps$, so the readbuffer provides a significant speedup. The ratio of the 99.9% percentile to the mean value can be considered as a crude estimate for the sensitivity to intra-die variatons. For the setup with readbuffer, this ratio is $\frac{delay_{99.9\%}}{delay_{50\%}} = 126.5\%$, while it is 143.3% for the setup without readbuffer. The setup with readbuffer obviously suffers less from intra-die variations. Figure 6.7 also shows the normal distributions that have been fitted to both full datasets. Notice that the tail of the distribution is not well predicted by such a normal distribution. This

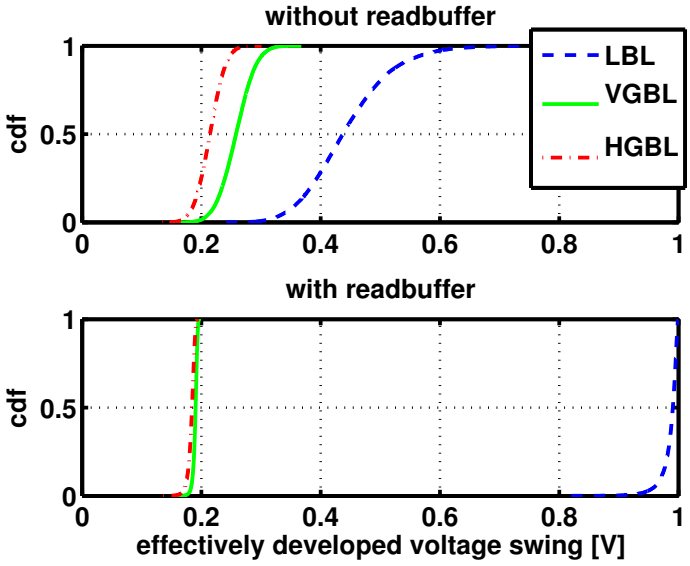


Figure 6.8: Distributions of the effectively developed swings on the different bit lines at the time when 99.9% of the cells have developed a swing of $150mV$ or more on $HGBL$. In the text, this is called the idealised stop time.

has been discussed in more detail in [Doo08]. Also notice that safe operation requires a much higher percentile, e.g. 1 failure in 10^9 rather than 1 failure in 10^3, so the overhead due to intra-die variations will be twice as big as those numbers suggest (about 6σ rather than about 3σ).

Fig. 6.8 shows the cumulative distribution of the effectively developed swings on the different bit lines at the time when 99.9% of the cells have developed a swing of $150mV$ or more on the $HGBL$. In the assumption that the associated yield is acceptable, this would be the ideal moment to stop the discharging of the GBLs. In reality, it is very hard to control this stop time accurately, and the GBL discharge will continue for quite a while after this ideal stop time, wasting energy for no good reason.

In the setup without readbuffer, the average discharge of $HGBL$ and $VGBL$ is $235mV$, significantly larger than the $150mV$ which was actually required for the SA. The average swing on the LBL is almost $450mV$, significantly larger than the swing on the GBLs.

For the setup with readbuffer, the spread on the swing is much smaller, and the main reason for this is simple: because $HGBL$ and $VGBL$ were precharged at only $0.2V$, not even the fastest path can generate a swing larger than this $0.2V$. The LBL always makes a full-swing transition, as was intended. With this bit line architecture, there is no need to control the end of the GBL discharge period accurately. With idealised timing fixed at the 99.9% delay percentile,

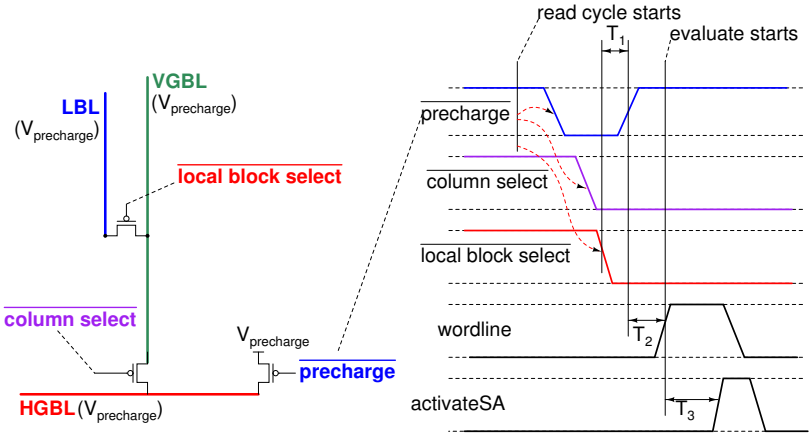


Figure 6.9: The precharge scheme used in [Yan05]. $VGBL$ and LBL are precharged by the $HGBL$ precharge transistor through the read pass transistors.

energy consumption for both schemes is about equal in this setup. With real timing or in a larger matrix, the advantages of the read buffer setup become more pronounced.

6.2.4 Precharge control

For the read operation, LBL, $VGBL$ and $HGBL$ must be precharged before the evaluation phase starts. This can be accomplished in many different ways. In this section, three example precharge control schemes will be described.

Figure 6.9 shows a precharge scheme in which $VGBL$ and LBL are precharged by the $HGBL$ precharge transistors through the pass transistors that are also used for the read operation. This scheme was used in [Yan05]. In this scheme, precharge of $VGBL$ and LBL can only take place when the decoder has already decoded which column and which local block is accessed. T_1, the overlap time between the precharge pulse and the local block activation should be large enough to precharge LBL. The word line should not be activated while the precharge transistor is still enabled, as this would result in wasted short circuit currents, hence T_2 should not be negative. The global read SA can only be activated after a time T_3 that the cell needs to develop a sufficient signal on $HGBL$. The timing constraint T_1 is the main disadvantage of this scheme. It increases the memory delay and it increases the total energy that is required for timing control. This scheme has two main advantages. First, all bit lines are floating almost all the time, which reduces the leakage currents. Second, no dedicated precharge transistors need to be placed and controlled. This reduces the area overhead, the bit line capacitance and the dynamic control energy. A reasonable estimate for the energy saving that can be obtained this way is 1 to $2fJ$ per bit in a $90nm$ technology with $Vdd = 1V$.

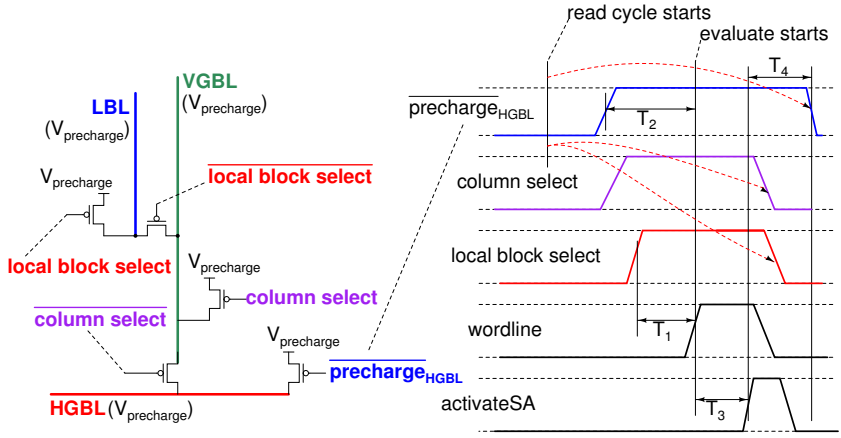


Figure 6.10: $HGBL$, $VGBL$ and LBL are precharged by their own dedicated precharge transistor. Precharge for the next access starts at the end of the current access, as soon as the specific bit line is no longer needed for the current access.

In the first scheme, all bit lines are precharged during the first phase of the read operation. In the second scheme that is shown in figure 6.10, all bit lines are precharged for the next access immediately after the current access is finished. This scheme requires dedicated access transistors on all bit lines, but the requirement for overlap between precharge pulse and local block activation is avoided, which simplifies timing and improves access delay. All local bit lines are precharged almost all of the time, which increases leakage somewhat. However, this scheme introduces two new issues. First, the HGBL should not be precharged before the read SA has finished its sensing operation, so T_4 must be large enough. In practice, the result of this constraint will often be that the actual activation edge of $\overline{precharge}_{HGBL}$ will occur in the next clock cycle, even though its timing is controlled starting from the current clock edge. This is unconventional, but it is not necessarily a problem. The second issue is more fundamental. Because the $HGBL$ precharge is activated before it is known whether the next operation will be a read or write operation, $HGBL$ will be precharged unnecessarily during write operations.

The third precharge scheme, shown in figure 6.11 solves these two issues. The precharge of $VGBL$ and LBL starts at the end of the previous access, while the precharge of $HGBL$ starts at the beginning of the new access. During write accesses, the precharge of $HGBL$ can be avoided.

6.3 Write

Figure 6.12 shows the traditional write scheme. The bit lines DL and $\overline{DL}$ are precharged to V_{prech}, which is typically around Vdd. For cells that must be overwritten, the appropriate bit line is pulled down to V_{low} by its write driver.

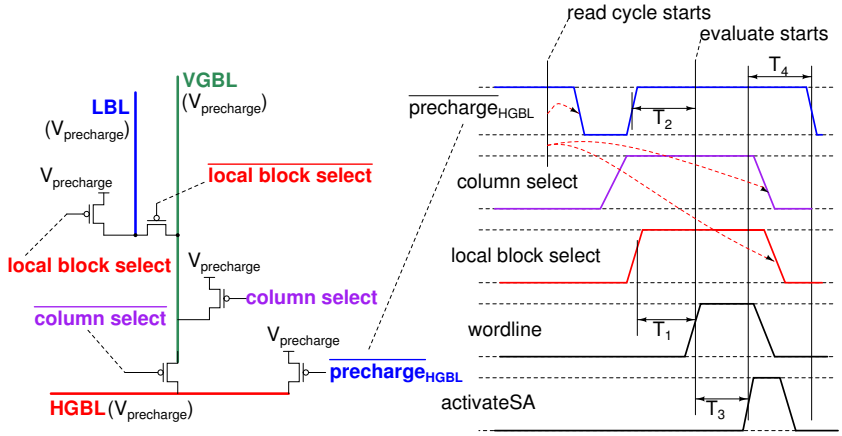


Figure 6.11: $HGBL$, $VGBL$ and LBL are precharged by their own dedicated precharge transistor. Precharge for the next access starts at the end of the current access for LBL and $VGBL$, but $HGBL$ is precharged at the beginning of the next cycle if it is a read cycle.

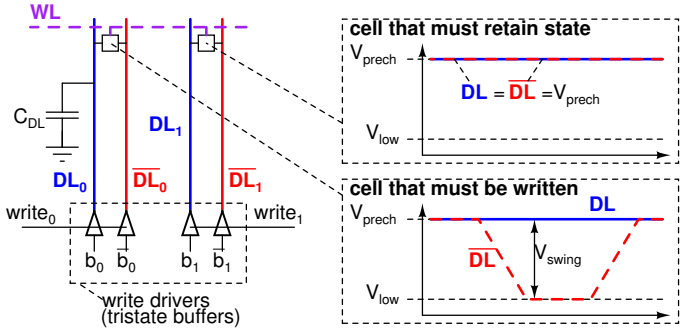


Figure 6.12: Traditional write scheme.

Typically, $V_{low} = 0V$. In many memory organisations, not all cells on the activated word line should be overwritten, either because the word line is not fully subdivided or because write masking is used, which means that only a selected subset of the bits in a word are altered. This selective write can be easily implemented in the traditional scheme: the activated cell in a column for which the write drivers are not activated, is stressed as in a normal read operation and hence retains its data.

The main disadvantage of the traditional write scheme is that it consumes a huge amount of energy as each bit requires one full-swing transition on a bit line.

$$E_{write,traditional} = C_{DL} \cdot Vdd^2$$

For example, in $90nm$, with $R = 512$ and $C = 256$ and subdivided bit lines, $C_{DL} \approx 200fF$. At $Vdd = 1V$, the energy consumption per bit is $200fJ$. For memories that operate in a single cycle, the read delay is typically larger than the write delay, so the write delay is not considered here.

Several techniques to reduce the write bit line energy will now be discussed: reducing the bit line swing, using another signalling scheme and charge recycling.

6.3.1 Reduced bit line swing

Several methods to reduce the write energy by reducing the bit line swing have been proposed. However, these methods reduce the operation margins for the cell, which limits their applicability in DDSM technologies. A good overview and discussion is provided in [Yan05]. This paper also introduces an elegant solution to reduce the write energy without reducing the cell margins, as discussed in section 3.5.4 and figure 3.19. Low-swing signals are used on the global bit lines. These low-swing signals are amplified by local sense amplifiers onto the local bit lines. Such local amplifiers for the write signals are called write receivers in this text. Although there is some implementation freedom, a reasonable value for the minimal required input voltage difference for a write receiver in $90nm$ is $\Delta V_{min,WR} = n \cdot \frac{A_{\Delta V_T}}{\sqrt{W \cdot L}} \approx 150mV$. The energy consumption is then

$$\begin{aligned} E_{write,writeReceiver,Vdd} &= E_{GBL} + E_{LBL} \\ &= C_{GBL} \cdot \Delta V_{min,WR} \cdot Vdd + C_{LBL} \cdot Vdd^2 \end{aligned}$$

For example, in $90nm$, with $R = 512$, $C = 256$ and $T = 16$, C_{GBL} is approximately $200fF$ and C_{LBL} is about $10fF$. At $Vdd = 1V$ and with a complementary voltage swing of $150mV$, the energy consumption per bit is $E_{LBL} = 10fJ$ and $E_{GBL} = 30fJ$, a reduction with a factor of 5 compared to the traditional scheme.

In [Cos07], we improved this scheme by sharing one write receiver between multiple local read bit lines, as shown in figure 6.13. This way, the different requirements of read and write can be addressed in a more independent way. For the write operation, a larger granularity is preferable, as the area overhead of a good write receiver is rather large and the local bit line is not dominant in

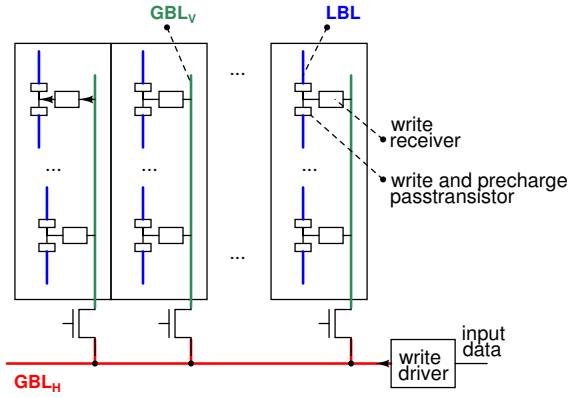


Figure 6.13: Shared write receivers. Each LBL has its own readbuffer.

the write energy. When a short buffered bit line is used for the read operation, the local bit line capacitance has an important impact on the read delay and the read energy, hence short read bit lines are desirable.

Write schemes with local amplification require local bit lines. Hence, the use of local bit lines should be considered, even in situations where they do not improve the performance of the read operation. In write schemes with local amplification, fine-grained write masking is more costly to implement, as the activation of the local amplifier must be blocked explicitly.

6.3.2 Signalling scheme

The traditional write scheme employs precharged complementary global bit lines. Alternative options are to use static signals (e.g. [Sil98]), or to use single-ended write signals. The energy consumption of the different signalling schemes is shown in figure 6.14.

With complementary static signals, energy is only consumed when the data to be written changes from 0 to 1 or from 1 to 0. p_0 is the probability that a "0" must be written and p_1 is the probability that a "1" must be written. In this text, it is assumed that the probabilities are independent of the position of the bit in the word and in the memory and also of the previous value that was written to the same column. The average write energy per bit reduces to

$$E_{write,static\ complementary} = (p_0 \cdot p_1 + p_1 \cdot p_0) \cdot C_{DL} \cdot Vdd^2$$

The relative energy reduces from 100% in the traditional scheme down to $(2 \cdot p_0 \cdot (1 - p_0))$, as shown in figure 6.14. Even with $p_0 = p_1 = 50\%$, the energy is reduced by a factor of 2.

In traditional memories, attempts to use static write signals on the global bit lines are generally not very effective. The global bit lines are shared for read and write operations, and two write operations are often separated by one or

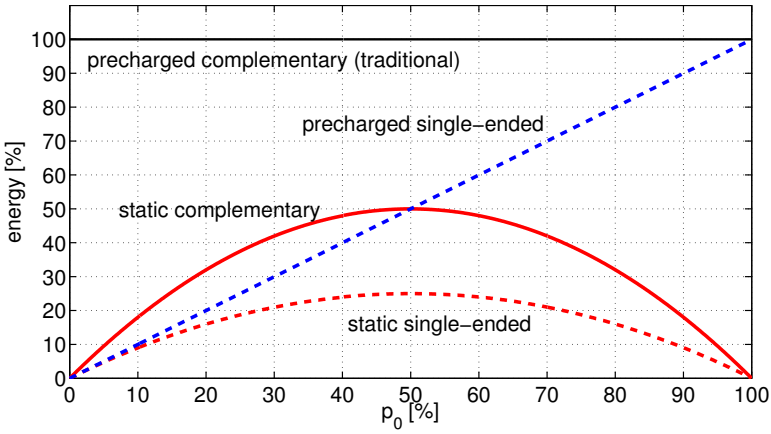


Figure 6.14: Energy consumption of different signalling types, relative to the energy consumption of a traditional complementary precharged write scheme. p_0 is the probability that a "0" must be written. It is assumed that this probability is independent of the position of the bit in the word and in the memory and of the previous value that was written to the same column.

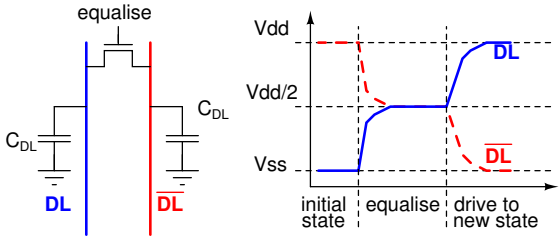


Figure 6.15: Charge recycling: an example.

more read operations which need to precharge the bit lines. Static write signals can be used when separate (global) write bit lines are used as in [Sil98] or when write operations are clustered.

It is possible to use single-ended signalling on the global bit lines, but the local bit lines should be complementary to avoid cell margin reduction, so a local inversion must be provided. This functionality could be integrated in a local low-swing write receiver. The relative energy consumption of precharged single-ended global bit lines and of static single-ended global bit lines is shown in figure 6.14 and is

$$E_{write,precharged\ single-ended} = p_0 \cdot C_{DL} \cdot Vdd^2$$

$$E_{write,static\ single-ended} = (p_0 \cdot p_1) \cdot C_{DL} \cdot Vdd^2$$

6.3.3 Charge recycling

Charge recycling can also reduce the energy consumption. Figure 6.15 shows a very simple example of charge recycling with complementary data lines. When the values on the data lines must be toggled, a traditional approach consumes

$$E_{toggle,traditional} = C_{DL} \cdot Vdd^2$$

With charge recycling, the two bit lines are first equalised with a pass transistor. During this phase, they both evolve towards $\frac{Vdd}{2}$ without consuming charge from Vdd. When equalisation is complete, the pass transistor is disabled and the data lines are driven to their final state. The data line that is pulled up draws a charge $Q = C_{DL} \cdot \frac{Vdd}{2}$ from Vdd, hence the total toggle energy is

$$E_{toggle,charge\ recycling} = C_{DL} \cdot \frac{Vdd}{2} \cdot Vdd$$

In this example, the energy consumption is halved. A similar scheme was used in [Mai98], but combined with a $\frac{Vdd}{2}$ precharge scheme, which results in a maximal reduction with a factor of 4, but which reduces the cell margin. [Kim08] introduces a much more involved charge recycling scheme. However, this scheme requires clustered write operations and significantly reduces the cell margin.

One of the main problems with all charge recycling schemes is that they require the use of pass transistors, typically with drain and source at $\frac{Vdd}{2}$. The pass transistor hence operates at reduced V_{gs}. Unless very low threshold transistors are available, the switch resistance of both NMOS an PMOS pass transistors is very large in this region. In some cases, it is easier to equalise the data lines to a separate supply. Because $Q_{high} = Q_{low}$, the current from this supply is small.

6.4 Summary

This chapter addresses the transfer of the data bits between the memory cell and the memory output interface during read and from the memory input interface to the memory cell during write.

Section 6.2 discusses the read operation. First, a comparison between complementary and single-ended sensing is provided. If p_0 is around 50%, the two schemes are rather comparable, both in terms of delay and energy, and the optimal trade-off depends on low-level implementation details. When p_0 is significantly below 50%, the advantages of single-ended signalling become very pronounced.

Second, a number of practical implementations are presented and their strong and weak points are highlighted. The traditional implementation (solution A) without bit line subdivision is outlined and used as baseline for comparisons. The main problems of this implementation are the large bit line capacitance and the fact that the bit line swing must be controlled by the word line pulse width, which in practice results in a larger swing than what is actually needed, and

hence a larger energy consumption. Another fundamental problem is related to the write operation: without local bit line, the low-power write schemes that use local write receivers cannot be used. Subdividing the bit line with pass transistors (solution B) improves the bit line energy and delay in typical memory organisations by about 25%. If the memory organisation does not require horizontal routing, the improvement increases to 50%. The short buffered local bit line with extended low-swing global bit lines scheme (solution C) that we introduced in [Cos07, Cos09] allows to use NMOS pass transistors rather than complementary pass transistors. The swing on the global bit line is automatically limited, without relying on accurate word line pulse width control. The low precharge voltage on the global bit lines allows for the use of a DC-DC converter or separate supply voltage to reduce the GBL energy consumption with an additional factor of 5. This has not been addressed in our work though. Section 6.2.3 clearly shows that solution C is faster and suffers less from intra-die variations than solution B for a $64KByte$ memory with $V_{swing} = 150mV$. As discussed in chapter 3, the short buffered bit line in solution C also improves the dynamic cell stability. Adding a local sense amplifier to the local bit lines (solution D) seems an interesting approach for future work. Adding a repeater SA between $VGBL$ and $HGBL$ (solution E and F) only seems interesting for larger memories or in cases where pass transistors would limit voltage scaling.

Section 6.3 addresses the write operation. The traditional setup is outlined. The bit line write energy in the traditional scheme is 5 to 20 times larger than the bit line read energy.

Several techniques to reduce the write energy are discussed. First, techniques that use reduced swing signals on the global bit lines are discussed. The most interesting approach is to add local low-swing write receivers that amplify the small-swing signal from the global bit lines to a full-swing signal on the local bit lines, as was first discussed in [Yan05]. In [Cos07], we improved this scheme by sharing one write receiver between multiple local read bit lines to allow separate optimisation of the granularity for read and write bit line periphery. With typical write receivers in $90nm$ technology, the complementary swing can be reduced to below $200mV$. At a supply voltage of $1V$, this reduces the write energy with a factor of 5. Other schemes that can be effective to reduce the bit line write energy are described as well: the use of static signalling, single-ended signalling and charge recycling.

Contributions and highlights

This work introduces a new bit line structure for the read operation: a short buffered bit line with a low-swing extended global bit line. This structure reduces the impact of the cell read current on the memory delay. This allows for a cell with higher threshold transistors and improves the resilience against variations in the cell transistors. With this structure, the bit line swing is well controlled by the initial precharge voltage, so no accurate timing control is needed to limit the bit line energy consumption. As the cell speed variations

become larger, this becomes a significant advantage. If in future solutions the low precharge voltage is generated in a more efficient way, the energy reduction of this scheme becomes very pronounced.

The most efficient way to reduce the write energy is the use of local amplifiers which amplify a low-swing signal global bit lines signal onto the local bit lines. This is a very strong argument to introduce local bit lines, even in relatively small memories.

Chapter 7

Global control circuits: decoder and timing

7.1 Introduction

This chapter covers the global control circuits. This includes the circuits for timing control and the decoder.

Section 7.2 discusses the decoder, the circuit that decodes the address bits and selects the corresponding block and word. Different organisations for the global decoder are compared and the implementation of the predecoders is illustrated.

Section 7.3 treats the timing control. During a memory access, several timing events occur, such as the activation of the word line and the activation of the global bit line precharge. To ensure correct operation, these events must satisfy several delay constraints. This section starts with an illustration of the timing events and their delay constraints for a typical low power memory. Many different approaches can be used to ensure the correct order of these events, so several choices must be made. These choices are outlined in the remainder of this section. First, timed delays can be generated by local timing circuits, or they can be generated by global timing circuits, in which case they still need to be transmitted to the correct local block. Second, more or fewer external clock events can be used as inputs for the timing control circuits. Third, three methods can be used to ensure that the correct operation order between events is respected: the simulator-based approach, the replica-based approach and the calibration-based approach.

A large part of the energy for the decoder and timing is associated with the transmission of signals across the memory matrix. Section 7.4 presents several schemes for the reduced-swing transmission of activation and selection signals.

Section 7.5 discusses the integrated timing and decoder solution that is used in our second prototype [Cos09].

Section 7.6 summarizes this chapter.

7.2 Decoder

The decoder is the circuit that decodes the address bits and selects the corresponding word by asserting its word line. The word line driver is discussed in section 3.3.

In most memories, additional select signals are required. When multiple words are located on a single row, a column select signal (CS) is required. In memories with subdivided word lines, CS must be transmitted to the local word line driver. In memories without subdivided word lines, where all words on the activated row are enabled, it suffices to transmit CS to the bit line periphery. In memories with both subdivided bit lines and subdivided word lines, a local block select signal ($activateBlock$) is required to control the precharge and operation of the local bit lines.

7.2.1 Decoder topologies

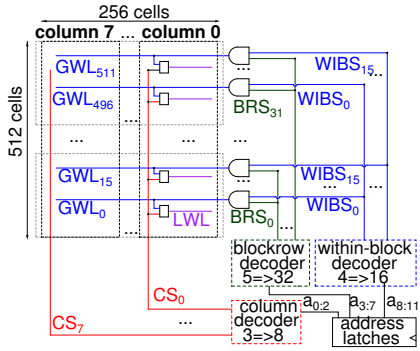
Figure 7.1 shows several global decoder topologies for a $128Kbit$ memory with a word length of 32 bits, organised as 256 cell columns by 512 cell rows. The height of the local block is $T = 16$ cells. The first layer of the decoder consists of the predecoders. In this example, these are the blockrow decoder, the within-block decoder and column decoder. Each of them decodes a subset of the address bits into a one-hot set of output wires. For example, the blockrow decoder decodes 5 address bits (a_3 to a_7) and asserts one out of $2^5 = 32$ blockrow select signals (BRS). The designer can decide freely which address bits to assign to which decoder. This choice can have a large impact on the activity of the

Table 7.1 tabulates the capacitance values for the wires in the different topologies. These capacitance values only include the wire parasitics and the input gate capacitances. Drain capacitances and capacitances inside decoder gates are not included here as they have little impact on the energy comparison that is made in this section. The output drivers of the gates do add significantly to the total energy consumption, but this energy consumption is almost proportional to the output load. The only significant effect of these gates on the conclusions is hence an increase of the total energy consumption with 15% for rather slow drivers and up to 50% for fast drivers [Wan05a].

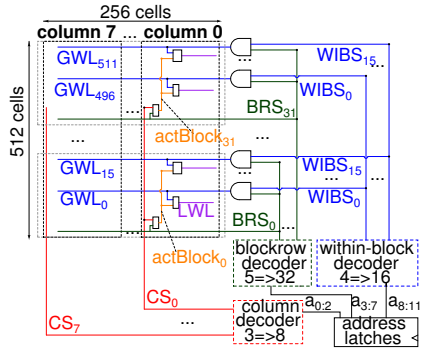
The average amount of switched capacitance is estimated for two very simple access patterns. In the linear access pattern, the address increments with one for each access. In the random access pattern, all address bits change at random, with $P[a_i = 0] = 50\%$. The switched capacitance multiplied with Vdd^2 is a good indication of the dynamic energy consumption. The values in the table are for a $128Kbit$ memory with a word length L of 32 bits, organised as 256 cell columns by 512 cell rows in a $90nm$ technology. An input gate capacitance C_G of $1fF$ is assumed. The wire parasitics are $C_H = 0.3fF$ per cell and $C_V = 0.15fF$ per cell.

The global word line GWL is not included in table 7.1. The capacitance and the activity of this signal are identical for all considered topologies:

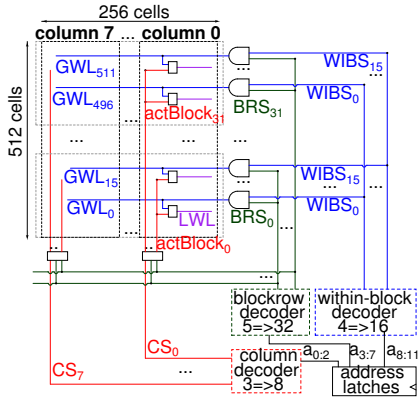
$$\begin{aligned} C_{GWL} &= (C - L) \cdot C_H + 0 \cdot C_V + \frac{C}{L} \cdot C_G \\ \text{e.g.} \quad & 67.2fF + 0fF + 8fF = 75.2fF \end{aligned}$$



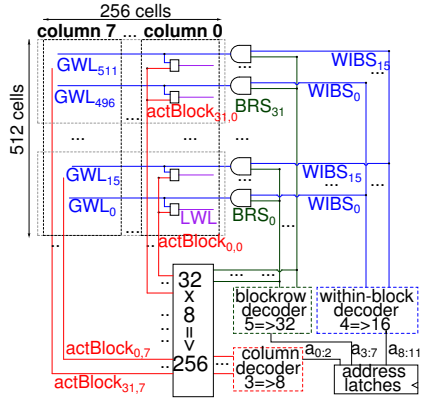
(a) Topology A (no *activateBlock*)



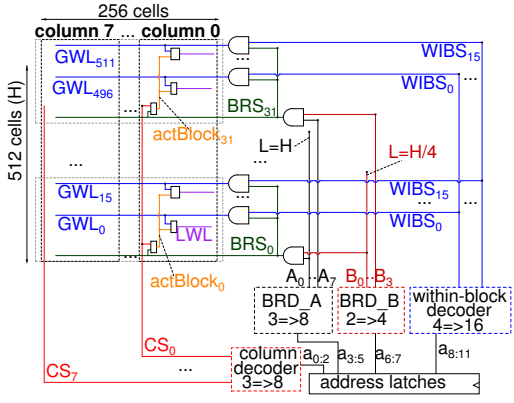
(b) Topology B (*activateBlock* generated in local block)



(c) Topology C (*activateBlock* generated in column footer)



(d) Topology D (*activateBlock* generated at predecoders)



(e) Topology E (multilevel blockrow decoder)

Figure 7.1: Example decoder topologies for a $128Kbit$ memory with a word length of 32 bits, organised as 256 cell columns by 512 cell rows.

	average wire capacitance	activity linear access	activity random access
CS	$\frac{L+(C-L)}{2} \cdot C_H + R \cdot C_V + R \cdot C_G$	100.00%	87.50%
	$38.4fF + 76.8fF + 512.0fF$	$627fF$	$549fF$
BRS	$0 \cdot C_H + \frac{T+R}{2} \cdot C_V + T \cdot C_G$	12.50%	96.88%
	$0.0fF + 39.6fF + 16.0fF$	$7fF$	$54fF$
WIBS	$0 \cdot C_H + \frac{(R-T)+R}{2} \cdot C_V + \frac{R}{T} \cdot C_G$	0.39%	93.75%
	$0.0fF + 75.6fF + 32.0fF$	$0fF$	$101fF$
topology A: average switched capacitance:		$635fF$	$704fF$
CS	$\frac{L+(C-L)}{2} \cdot C_H + (R-T) \cdot C_V + \frac{R}{T} \cdot C_G$	100.00%	87.50%
	$38.4fF + 74.4fF + 32.0fF$	$145fF$	$127fF$
BRS	$(C-L) \cdot C_H + \frac{T+R}{2} \cdot C_V + (T + \frac{C}{L}) \cdot C_G$	12.50%	96.88%
	$67.2fF + 39.6fF + 24.0fF$	$16fF$	$127fF$
WIBS	$0 \cdot C_H + \frac{(R-T)+R}{2} \cdot C_V + \frac{R}{T} \cdot C_G$	0.39%	93.75%
	$0.0fF + 75.6fF + 32.0fF$	$0fF$	$101fF$
actBlock	$0 \cdot C_H + T \cdot C_V + T \cdot C_G$	100.00%	99.61%
	$0.0fF + 2.4fF + 16.0fF$	$18fF$	$18fF$
topology B: average switched capacitance:		$180fF$	$373fF$
CS	$\frac{L+(C-L)}{2} \cdot C_H + 0 \cdot C_V + \frac{R}{T} \cdot C_G$	100.00%	87.50%
	$38.4fF + 0.0fF + 32.0fF$	$70fF$	$62fF$
BRS	$(C-L) \cdot C_H + \frac{T+R}{2} \cdot C_V + (T + \frac{C}{L}) \cdot C_G$	12.50%	96.88%
	$67.2fF + 39.6fF + 24.0fF$	$16fF$	$127fF$
WIBS	$0 \cdot C_H + \frac{(R-T)+R}{2} \cdot C_V + \frac{R}{T} \cdot C_G$	0.39%	93.75%
	$0.0fF + 75.6fF + 32.0fF$	$0fF$	$101fF$
actBlock	$0 \cdot C_H + \frac{T+R}{2} \cdot C_V + T \cdot C_G$	100.00%	99.61%
	$0.0fF + 39.6fF + 16.0fF$	$56fF$	$55fF$
topology C: average switched capacitance:		$143fF$	$345fF$
CS	$\frac{L+(C-L)}{2} \cdot C_H + (R-T) \cdot C_V + \frac{R}{T} \cdot C_G$	100.00%	87.50%
	$38.4fF + 74.4fF + 32.0fF$	$145fF$	$127fF$
A	$0 \cdot C_H + \frac{(R-T)+(R-\#B \cdot T)}{2} \cdot C_V + \#B \cdot C_G$	12.50%	87.50%
	$0.0fF + 70.8fF + 4.0fF$	$9fF$	$65fF$
B	$0 \cdot C_H + \frac{(R-T)+((\#A-1) \cdot T)}{2} \cdot C_V + \#A \cdot C_G$	1.56%	75.00%
	$0.0fF + 45.6fF + 8.0fF$	$1fF$	$40fF$
BRS	$(C-L) \cdot C_H + T \cdot C_V + (T + \frac{C}{L}) \cdot C_G$	12.50%	96.88%
	$67.2fF + 2.4fF + 24.0fF$	$12fF$	$91fF$
WIBS	$0 \cdot C_H + \frac{(R-T)+R}{2} \cdot C_V + \frac{R}{T} \cdot C_G$	0.39%	93.75%
	$0.0fF + 75.6fF + 32.0fF$	$0fF$	$101fF$
actBlock	$0 \cdot C_H + T \cdot C_V + T \cdot C_G$	100.00%	99.61%
	$0.0fF + 2.4fF + 16.0fF$	$18fF$	$18fF$
topology E: average switched capacitance:		$186fF$	$442fF$

Table 7.1: Capacitance values for the wires in the different topologies from figure 7.1. ($\#A = 8$, $\#B = 4$, $T = 16$, $R = 512$, $C = 256$, $L = 32$, $C_V = 0.15fF$, $C_H = 0.3fF$, $C_G = 1fF$). The global word line capacitance is not included.

$$\begin{aligned} \text{activity}_{\text{linear access}} &= (\frac{1}{2})^3 = 12.5\% \\ \text{activity}_{\text{random access}} &= 1 - (\frac{1}{2})^9 = 99.8\% \end{aligned}$$

The GWL hence adds between $75fF$ and $9.4fF$ to the average switched capacitance of all the topologies.

Topology A (figure 7.1a) does not provide a block activation signal, hence it cannot be used for memories with subdivided bit lines. Even for memories without subdivided bit lines, it is not a very good choice. The column select signal CS_i connects to all 512 word line drivers in a column. This results in a huge capacitive load on this wire (table 7.1).

The most efficient way to reduce the number of gates that connect to CS_i is to use the BRS signals not only for the generation of the horizontal select lines (BRS and $WIBS$ result in GWL), but also for the generation of the vertical select lines (BRS and CS result in *activateBlock*, which is abbreviated as *actBlock* in the figures). All topologies except A in figure 7.1 use BRS twice. In topology B (figure 7.1b, the BRS signals are transmitted over the appropriate column. Inside the locale block, an AND-gate combines BRS and CS to form *activateBlock*. This topology hence provides two advantages: it provides a block activation signal, which is needed to control the local bit lines, and it reduces the total amount of capacitance that is switched (table 7.1). The additional decode stage does not significantly increase the memory access time, as the decode path through GWL (which also occurs in topology A) is equally deep.

Topology C (figure 7.1c) is very similar to topology B, but now all BRS signals are transmitted at the bottom of the memory. In each column, they are combined with the corresponding CS_i signal. This generates the *activateBlock* signals for that column, which are transmitted to the correct local block. The average amount of switched capacitance is slightly lower in topology C than in topology B (table 7.1) because *activateBlock* is only transmitted over the height that is strictly needed, while CS in topology B is always transmitted over the entire height of the column. Topology C has some potential disadvantages. First, 32 *activateBlock* signals must now be routed over each column, as compared to one CS signal in topology B. Second, All 32 BRS signals must now be routed at the bottom of the memory matrix. This will most likely require some additional area and will hence increase the capacitance of all other wires in the memory. In topology B, it is probably possible to route one BRS wire per block row in the space that is already available. Third, when timing circuitry is added inside the local block to skew the edges of *activateBlock* as in figure 7.8, the *activateBlock* signal must be buffered a first time for the transmission from the column footer to the local block, and a second time after the skewing circuits to drive the actual gates. This will most likely increase the delay and energy consumption.

Topology D (figure 7.1d) shows what goes wrong when too much signals are combined too early in the decoder: too many wires must be routed in the available space, hence the area increases. The layout of the 8x32 to 256 decoder is also challenging.

In topology E (figure 7.1d), the combination of signals is delayed until further in the decoder: the blockrow decoder is distributed over the row headers. Predecoders generate $A_{0:7}$ and $B_{0:3}$. These signals are transmitted over the row headers. In each row header, the appropriate A and B are combined into the BRS signal. This topology increases the amount of switched capacitance and the decoder delay. The only advantage is that fewer wires must be routed over the row headers, $8+4$ compared to 32. Such a reduction can be useful in larger memories.

For typical memories, topology B and C are the most appropriate choices. The difference between both is subtle.

Table 7.1 clearly shows that the access pattern has a significant impact on the energy consumption in the decoder wires. For topology C, the energy consumption with a linear access pattern is 2.4 times smaller than for a random access pattern.

For all topologies except topology A, the wire capacitances contribute more to the load capacitances than the gates. In topology B, with the mentioned assumptions for gate and wire capacitance, the gate capacitance is only 25% to 30% of the total load capacitance. Hence, AND gates with an input capacitance $C_G = 1fF$ are probably a reasonable choice. For a normal CMOS implementation, this corresponds to an NMOS and PMOS with $W = 400nm$. AND gates with much larger input capacitances should be avoided. If there are compelling reasons to use such gates, a decoder topology with less gates connected to each wire should be considered. Such topologies can be obtained by decoding more address bits in more than one decode path, which is similar to the transition from topology A to topology B.

7.2.2 Predecoders

Figure 7.2 shows a schematic view of a $N = 3$ to $2^N = 8$ predecoder and the input latches for its address bits. Each of the address signals (a_0, $\overline{a_0}$, ...) connect to $2^{N-1} = 4$ NAND-gates with N inputs. For each applied address, exactly one of the NAND gates has a low output. In an implementation with static NAND gates, glitches occur when the address signals have 1-1 overlaps. The very worst case occurs when all address bits $a_{[0:N]}$ and $\overline{a_{[0:N]}}$ are 1 during the transition of address signals. In this case, $2^N - 1$ NAND gates generate a glitch which can propagate to the output drivers. This can easily be avoided by using dynamic gates or by using a latch design with a fast falling output and a slow rising output.

A rather large output driver is needed to drive the output signals onto the long wires that run across the memory matrix. To keep the wire load for the address

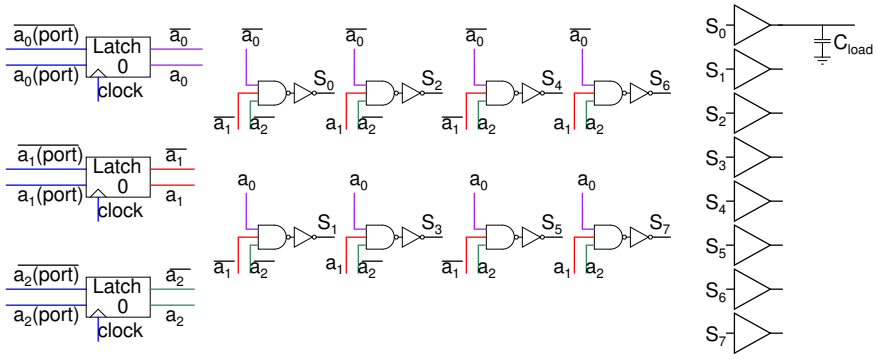


Figure 7.2: A simple 3 to 8 predecoder with input address latches.

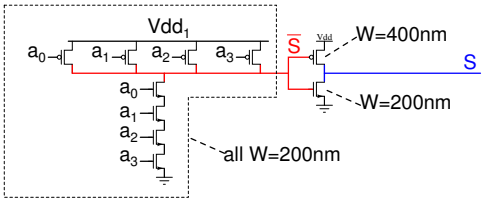


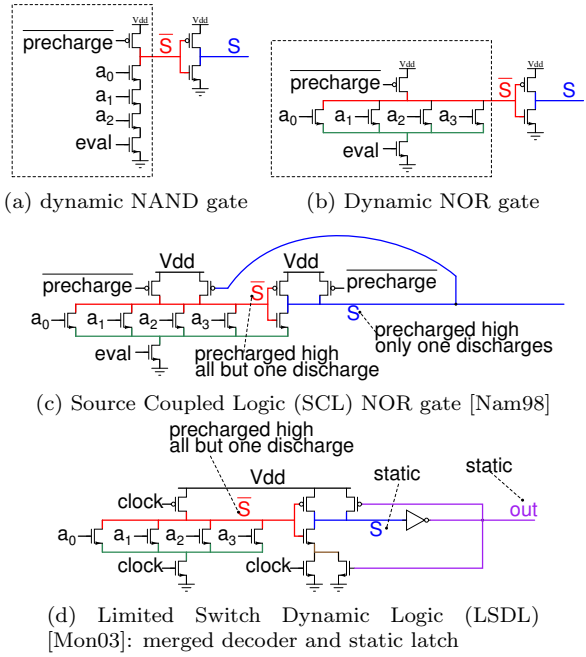
Figure 7.3: A 4-input static NAND gate.

signals as small as possible, it is better to move these drivers to the outside of the predecoders, as illustrated in figure 7.2.

When N is larger than 4, the wire and gate load on the address signals become too large, and the speed and energy performance of most N-input NAND-gates is poor. In such cases, an implementation with two layers of logic should be considered. AND-AND and NAND-NOR implementations both have their merits. An AND-AND implementation only employs the more efficient NAND gates and has an intrinsic buffer invertor, which is more suited to drive long wires and a high fan-out. A NAND-NOR implementation is more compact.

There are two interesting possible locations to place the second layer of such a two-layer implementation. Either clustered with the first stage, or distributed along the entire matrix edge, as in figure 7.1e.

Many different implementations have been proposed for the latches and gates. Figure 7.3 shows a trivial example: a 4-input static NAND gate. In $90nm$ at $Vdd = 1V$, with all SPSVT transistors, the nominal delay from the 50% crossing of the input signal to the 50% crossing of $\overline{S}$ for the worst-case address transition is $46ps$. This delay depends on the address transition. When a NAND-gate is de-activated, $4.2fC$ charge is taken from Vdd_1. For relatively large memories (e.g. $128Kbit$, 512 by 256 cells), the corresponding effective capacitance of $4.6fF$ is small compared to the load of the output wire, about 5 to 10 percent. A disadvantage of such static decoders is the fact that different address transitions result in different delays, which complicates timing.



(a) dynamic NAND gate (b) Dynamic NOR gate

(c) Source Coupled Logic (SCL) NOR gate [Nam98]

(d) Limited Switch Dynamic Logic (LSDL) [Mon03]: merged decoder and static latch

Figure 7.4: Implementations for predecoder gates.

For larger predecoders, the energy consumption associated with the address signals is significant as well. Each address signal connects to 2^{N-1} gates with an input capacitance of about $0.6fF$. Even without considering the wire capacitance, each address signal that changes, charges $4.8fF$.

Figure 7.4 shows several alternative implementations for predecoder gates. A dynamic NAND gate (figure 7.4a has a slightly lower internal capacitance and input capacitance for the address bits. Only one NAND gate from the predecoder is discharged each cycle. The output signal returns to 0 during the precharge phase, which changes its activity to 100% in the analysis from section 7.2.1. In practice, this implementation does not provide much benefits compared to a static NAND gate, especially when bleeders are required at the internal nodes to ensure signal integrity.

The dynamic NOR gate from figure 7.4b is much faster than the previous approaches. Its energy consumption is not acceptable though: all gates and all output wires except one make a transition during each cycle.

In [Nam98], the Source Coupled Logic (SCL) NOR gate (figure 7.4c) was introduced. This gate avoids the transitions on the output wires. All but one of the internal nodes of the gates discharge during each evaluation, but only one output wire makes a transition. This gate is a good solution for small predecoders when relatively high speed operation is required.

The gate that is depicted in figure 7.4d was introduced in [Mon03]. It has two interesting features. First, the output signal is static, which can reduce its activity. Second, this gate combines the functionality of the address input latches and the predecoder logic.

7.3 Timing

7.3.1 Events and their delay constraints

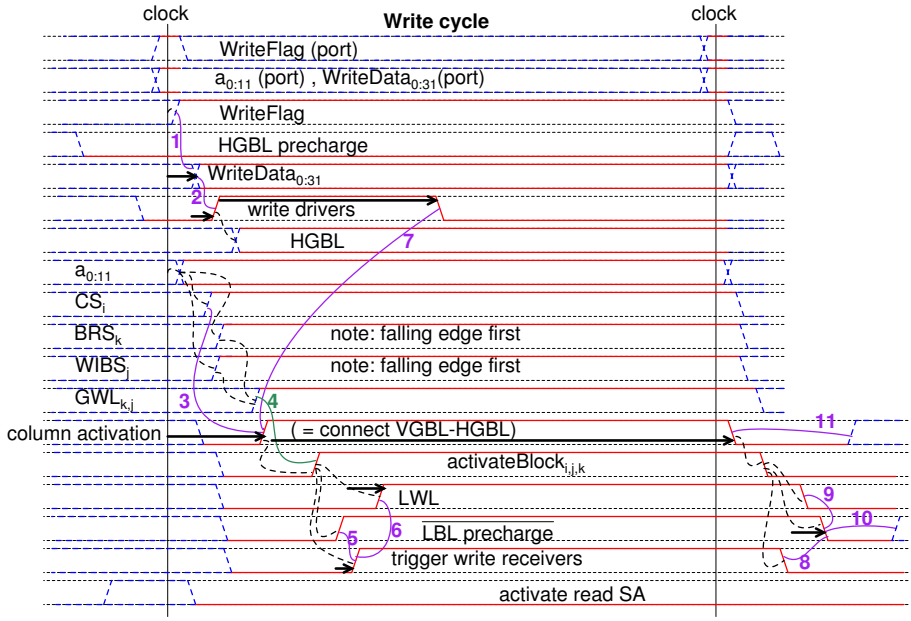
During a memory access, several timing events occur, such as the activation of the word line and the activation of the global bit line precharge. To ensure correct operation, these events must satisfy several delay constraints.

Figure 7.5 illustrates the important events during the write and the read operation for a typical low power memory. The important delay constraints are indicated. The number and criticality of constraints depend on the implementation of the other memory components. Especially the choice of the local bit line periphery (section 3.5) and of the global bit line solution for read, write and precharge (section 6.2.2, 6.2.4 and 6.3) has a significant impact. However, this example provides a good feel for the kind of delay constraints that must be satisfied by the timing circuits.

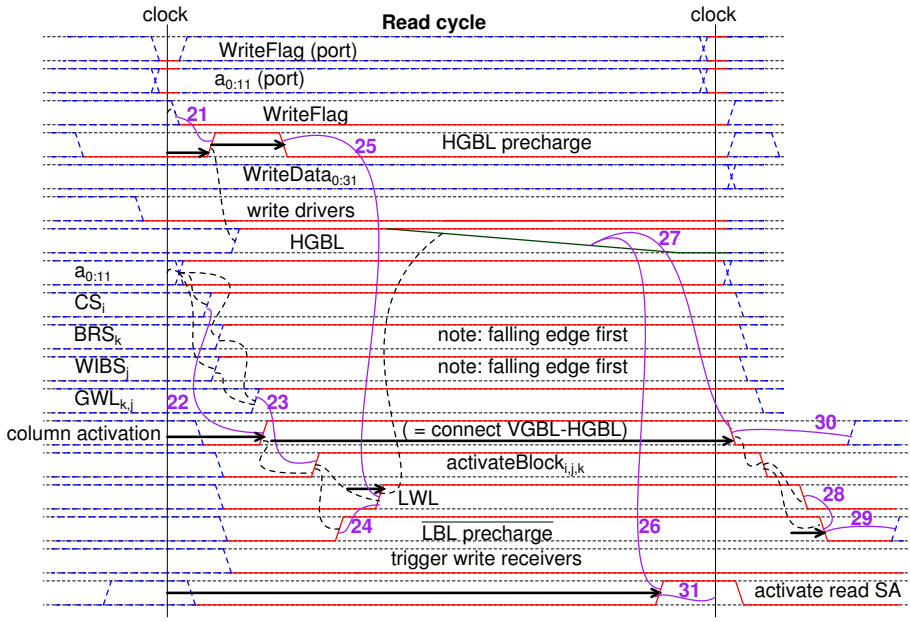
Figure 7.5a shows the events for the write operation. Before the clock edge, the input of the *writeFlag* latch must be high, and the inputs of the $writeData_{0:31}$ latches and $address_{0:11}$ must be ready. These inputs are indicated as (port) on the figure. The *address* and *writeFlag* latches are triggered by the clock. The *writeData* latches can also be triggered immediately by the clock, but as they are only required during the write cycle, they can also be clock gated with the *writeFlag* signal. Two options are available: either the clock is gated with the $writeFlag(port)$ input signal or the clock is gated with the latched *writeFlag* signal as in figure 7.18. The first option requires early arrival of the $writeFlag(port)$ signal, which increases its effective required setup time. The second option increases the required hold time of $writeData_{0:31}(port)$. In this second option, the clock for the *writeData* latches must be delayed until the output of *writeFlag* latch is ready, constraint (1) on the figure. When the write drivers are enabled, *writeData* is copied to the global bit lines. To avoid glitches, the write drivers should not be enabled before the *writeData* signals are stable (2).

When the address bits change, the predecoders that are implemented with static gates activate the appropriate select signals: CS_i, BRS_k and $WIBS_j$. The static global word line decoders then activate $GWL_{j,k}$.

Then the column is activated: the precharge of the $VGBL$s is disabled, the pass transistors between $VGBL$ and $HGBL$ are enabled and the $activateBlock_{i,k}$ signal is generated and transmitted to the local block. These events should only happen when CS_i is ready. This can be ensured in two ways: either by using a separate column activation signal which arrives later than CS_i (3) as



(a) write



(b) read

Figure 7.5: Important events for the memory described in [Cos09]. Delay constraints between events are indicated with solid curved lines and bold numbers. Natural consequences are indicated with dashed curved lines. Thick arrows indicate explicit delays that are added to satisfy the constraints.

in [Cos09] or by using a dynamic column decoder (or another decoder whose output returns to zero between two accesses), in which case CS_i can be used directly to control the column activation. To ensure correct write operation, there must be sufficient overlap between the write driver activation period and the period during which $VGBL$ is connected to $HGBL$ (7).

$activateBlock_{i,k}$ activates the local block: it disables the precharge of the local bit lines, it combines with GWL_j to activate $LWL_{i,j,k}$ and together with $writeFlag_i$, it controls the local write receivers. To avoid word line glitches and unintended timing skews, GWL must be settled before $activateBlock$ arrives (4). LWL should only be enabled when the local write receivers have already driven the local bit lines to the correct state (6), which can only start when the local bit line precharge is disabled (5).

During the deactivation, the order of events inside the local block should be the other way around: the bit line precharge should only be activated when the LWL and the write receivers have been disabled (8,9). The precharge of $VGBL$ should be completed before the next cycle activates the column (11), and LBL should be precharged again before the next cycle activates the local block (10).

Figure 7.5b shows the events for the read operation. After activation, the $HGBL$ is precharged. This precharge is only needed during the read operation and hence should only start after $writeFlag$ is latched (21). Again, column activation must happen after CS_i is available (22), and $activateBlock$ must arrive after GWL (23). To avoid static currents, LWL should only be activated after the LBL precharge is disabled (24). $HGBL$ precharge should already be disabled as well (25). The global read SAs must be activated after sufficient swing has developed on $HGBL$ (26) but sufficiently before the next clock edge to ensure timely output data (31). The connection between $VGBL$ and $HGBL$ can only be interrupted after sufficient swing has been developed on $HGBL$ (27). LWL should close before LBL precharge is enabled again (28). LBL precharge and $VGBL$ precharge must be enabled in time to ensure that the wires are precharged correctly for the next access (29,30).

Not all the delay constraints are equally critical. Some of the constraints introduce functional failures when they are violated. For example, if the global read SAs are activated before a sufficient voltage signal has been developed on $HGBL$, some SAs will give incorrect results. In most memories, this cannot be tolerated. For some other constraints, violation results in additional short circuit currents. This happens for example when LBL precharge is enabled before LWL is disabled. If the overlap is not too long or occurs only for a limited percentage of the words in the memory, the additional energy consumption can often be tolerated.

7.3.2 Local versus global timing control

This section will focus on the events that must occur inside the local block that is accessed, e.g. disabling LBL precharge, enabling the local word line,

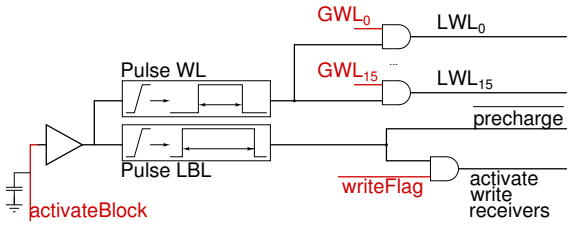


Figure 7.6: Schematic example of an entirely local timing circuit for the local block. The only external timing event is the rising edge of *activateBlock*. GWL and *writeFlag* must be stable before *activateBlock* arrives.

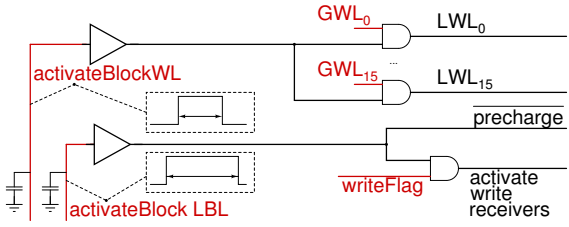


Figure 7.7: Schematic example of a very global timing circuit for the local block. Both *activateBlockWL* and *activateBlockLBL* are global signals. Both the rising and falling edge of these signals are used.

disabling the local word line, enabling LBL precharge. Similar considerations can be applied to the events that must occur in the accessed column, such as connecting $VGBL$ with $HGBL$, but these are not discussed here.

The initial trigger to activate a local block is always an external signal that is transmitted to the local block. This external signal is generated inside the global timing circuitry of the memory and transmitted only to the local block that is accessed, as indicated by the address bits.

In principle, all timing for the local block can be derived from a single external trigger signal by using local delay elements and pulse generators. Figure 7.6 provides a schematic view of such an entirely local timing circuit. The rising edge of *activateBlock* is the only external timing event that is used. The delay lines in the pulse generators must be comparable in delay to the cell read delay. For example: if a delay of $500ps$ is required at a power supply voltage of $1V$, a delay line with 40 SPSVT transistors or 25 SPHVT transistors is needed. Such an entirely local timing control circuit has some significant disadvantages. The timing circuits are duplicated in each local block. This introduces a significant area overhead and has an impact on the memory leakage. The timing circuits must be as small as possible, which limits the available design options. Because the memory only operates correctly if each of the instances of the timing circuit generates acceptable timing pulses, Such a fully local implementation further increases the importance of uncorrelated intra-die variations.

Figure 7.7 provides a schematic view of a timing circuit in which all timing

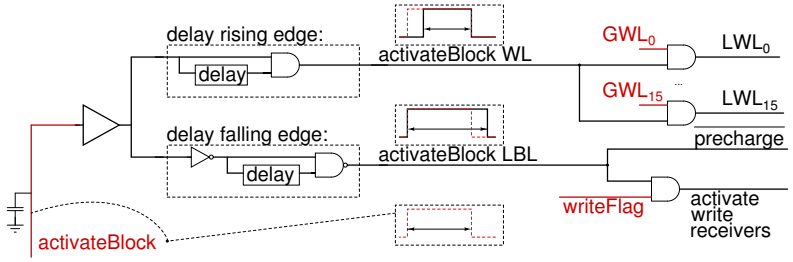


Figure 7.8: Schematic example of a timing solution for the local block which uses both global and local timing control.

generation has been moved to the global level. Here, two timing signals are sent to the local block, $activateBlockWL$ and $activateBlockLBL$, and both rising and falling edge of these signals are used to control the local block timing. In such an implementation, the area of the local periphery is very small. The area consumption of the global timing circuit is less critical, so more advanced timing circuits can be used to generate the global timing signals $activateBlockWL$ and $activateBlockLBL$. This also makes it easier to use different timing settings for different operation modes. Intra-die variations have less impact on the generation of timing pulses in such a global scheme, as the pulses for all local blocks are generated by a single circuit. However, during the transmission to the local block, significant skew between signal edges can still be introduced, both for different edges for the same local block and for corresponding edges for different local blocks. This can be a significant issue if the signal transition on the global timing wires is slow, as small differences in the threshold voltages of the receiver circuits in the local block result in large delay differences. The main disadvantage of such a very global implementation of timing control is that it requires more transitions on global wires with a heavy wire load, which adds to the dynamic energy consumption.

Figure 7.8 shows a schematic example of a timing solution for the local block that uses both global and local timing control. Only one global timing signal $activateBlock$ is used, but both the rising and falling edge of this signal are used. The pulse width of $activateBlock$ is defined by the global timing circuitry and defines the basic activation time of the local block. Local timing circuits are used to ensure the correct order of the local events: first disable precharge, then activate LWL, then deactivate LWL, finally enable precharge again. This is ensured by delaying the rising edge for the word line activation, and by delaying the falling edge for the bit line periphery activation. These delay skews are much smaller than the total pulse duration, hence the local timing circuit is smaller than in a fully local implementation while the amount of global signal transitions is reduced as compared to a fully global timing solution.

7.3.3 Reference external events for timing

For embedded SRAMS, the memory operation is triggered by the system clock. Pipelined memories use several consecutive clock edges to control the memory timing. This text only discusses memories which operate withing a single clock cycle. Such single-cycle memories can use at most three clock edges: the initial rising clock edge, the first falling clock edge and the next rising clock edge.

It is possible to use only the initial rising clock edge, as was done in both our designs [Cos07, Cos09]. This has two main advantages: the operation is not affected by the duty cycle of the clock and the memory can distribute the available clock period between the different operation phases in the most effective way. There are two main disadvantages though. First, delay elements with a delay up to the total clock period must be generated accurately, which can be difficult and which consumes energy. Second, the activation of the SA is very critical. It must happen as late as possible to maximise the time that is available for the rest of the memory, but it should still occur sufficiently before the next clock edge to ensure stable output data.

Some designs use one phase of the clock to precharge bit lines and decoder gates and use the other phase to actually access the memory, e.g. [Gee09]. This reduces the maximal delay that must be generated internally. However, if the duty cycle of the clock is not well matched with the delay requirements of the memory, time is wasted in the clock phase that is too long.

In some cases, the problem of accurately controlling the SA activation can be avoided by using the next rising clock edge as activation signal for the global read SAs. A SA that is followed by a set-reset latch can be used as an edge-triggered flip-flop [Nik00]. This flip-flop can be used as part of the memory data output register.

7.3.4 Methods to ensure correct operation order

7.3.4.1 Simulator-based timing

In simulator-based timing, a circuit-level simulator such as SPICE is used to verify the correct operation in all process corners. If timing violations are detected, they are fixed by inserting additional delay elements in some paths. These delay elements are typically invertors. In DDSM technologies, uncorrelated intra die variations must be considered in the verification phase.

This approach has a very significant disadvantage. The delay elements that are used do not have the same dependency on process parameters, temperature and supply voltage as the actual circuit that must be controlled. To ensure correct operation under all circumstances then requires the introduction of large delay margins.

However, this technique can be very useful in cases where only small delay differences must be created, as in figure 7.8.

7.3.4.2 Replica-based timing

In replica-based timing, the delay elements are replaced with replicas of the actual circuit that must be controlled. When process parameters, temperature and supply voltage deviate from their nominal values, the delay of such replicas accurately tracks the delay of the circuit that must be controlled. Replica based timing suffers from three problems. First, it is rather difficult to create a replica delay for the global bit line discharge, which is the delay that is needed to disable the word line and to activate the global read SAs. Typical solutions use a replica cell which is activated at the same time as the actual cells of the word. The replica cell discharges a replica bit line. The voltage swing on the replica bit line is monitored until it is sufficiently large to allow for the SA activation. The main problem when using an identical replica is that the swing on the replica bit line is a small-swing signal and a fast analog amplifier is needed to monitor the swing. In [Amr98], two interesting alternative options are described, which both rely on enlarging the swing on the dummy bit lines as compared to that on the actual bit lines. One option to enlarge the swing is to use replica bit lines with a smaller capacitance than that of the actual bit lines, the other options is to activate multiple replica cells on a single replica bit line.

Second,there is always a delay between the completion detection by the replica circuit and the actual action which is triggered by it. This can be compensated to some extent by increasing the speed of the replica circuit, but the larger the difference between actual and replica circuit is made, the more the solution evolves towards that of the simulator-based approach.

Third, replica-based timing has another, more fundamental problem when applied in DDSM technologies with large, uncorrelated intra-die variations. The basic operation principle of replica-based timing is to transform the timing problem from accurately controlling an absolute delay to accurately controlling the difference between two delays. This is great if the delay variations of the two paths are strongly correlated. However, in the case of uncorrelated variations, $\sigma_{\Delta delay} = \sqrt{2} \cdot \sigma_{delay}$. In this case, replica-based timing can actually make things worse.

This can be avoided to some extent. For example, in [Osa01], multiple replica cells are activated to reduce the impact of intra-die variations.

7.3.4.3 Calibration-based timing

Calibration-based timing makes use of programmable delay elements. The optimal settings for these programmable delay elements for the actual die and operation mode are obtained during a calibration phase. Section 7.5.2 discussed the programmable delay element that is used in [Cos09]. In [Suz02], a programmable delay is used for the SA activation. In [Kru08], software programmable pulse widths are introduced.

Calibrated timing has several important advantages. It can adapt perfectly to the transistor parameters of the actual die and to the selected operation

mode. The variations due to transistor mismatch in the programmable delay elements are taken into account during calibration, hence minimal sized delay elements can be used and the only margins that are needed are those against the variations of the circuits to be controlled. Compared to simulation-based timing approaches, calibrated timing reduces both the required design margins and the design risks.

The main disadvantage of calibrated timing is that a separate calibration phase must be provided and an appropriate calibration algorithm must be implemented. In [Lai09], the calibration algorithm was integrated into the build-in-self-test (BIST) circuit. The presented design uses a hybrid timing approach, in which a replica bit line is complemented with calibrated timing. The provided programmable delay element has only four settings, which cover only a small delay range.

7.4 Signal transmission

When global timing signals are used to control the behaviour of the local block that is being accessed, these signals must be guided to that local block by the decoder. Often, the functionality of decoder and timing control wires are merged to some extent. For example, when the column select signal CS is implemented as a return-zero signal, the delay and pulse width of CS can also be used to control the column and local block timing, the latter through its impact on *activateBlock*. If the most active wires (CS in table 7.1) are replaced with a return-zero signal, their activity hardly increases. This is hence a simple and efficient technique.

As the signal transitions on the global wires of the decoder and timing control typically dominate their energy consumption, it makes sense to use a signalling scheme with reduced swing on these wires.

For signals that only contain binary information, no timing information, a low-swing signalling scheme with a strobed sense amplifier as receiver can be used. The strobing timing is derived from an activation signal that was already needed anyway. This strobed receiver scheme is for example efficient to transmit the write flag to the local block. The signal that controls the LBL precharge can be used to trigger the receiver for the write flag. Only one SA must be triggered in this case.

A similar scheme with strobed sense amplifiers can be used to receive the GWL signal. However, in this case, several SAs must be triggered, as it is unknown which GWL will be activated. If T is 8, the net result might still be benificial, for larger values of T, the SA energy consumption becomes larger than that of the original full-swing GWL signal. With a SA per word line, area and leakage can prove problematic as well. Such strobed low-swing transmission of information was used in [Cos07].

The low-swing transmission of activation signals is more challenging, as the receiver must trigger based on the low-swing input itself.

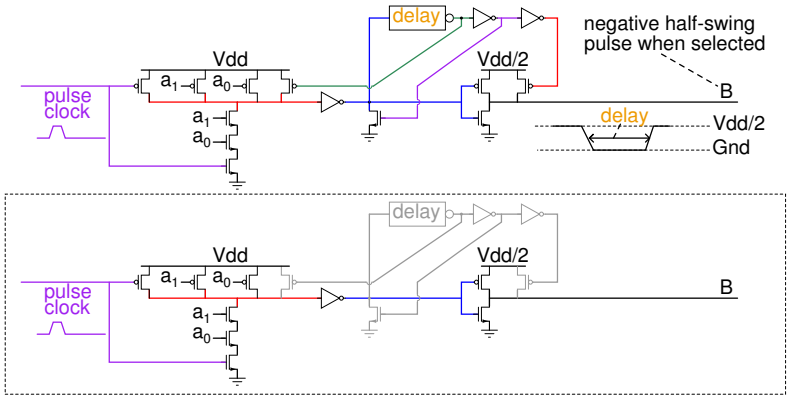


Figure 7.9: Predecoder gate with negative half-swing pulse output with self-timed reset [Mai98].

In [Mai98], a clever half-swing signalling technique was used. Figure 7.9 illustrates the predecoder gate with negative half-swing output B (from $Vdd/2$ to $0V$) that is used in this design. A similar predecoder generates a positive half-swing output A (from $Vdd/2$ to Vdd). Both predecoders generate pulsed outputs, which increases the time that is available to drive the output signal back to $Vdd/2$, which is a slow operation due to the reduced Vgs of the driver transistor. The predecoder transistors have a very low V_T of $0.1V$. The combination of the pulsed nature of the output signals with matched capacitance values ensures that very little net current is drawn from the $Vdd/2$ supply voltage.

Figure 7.10 shows the gate that is used at the receiving end. In [Mai98], a 3-input AND gate was used rather than the displayed 2-input version. The figure also shows the different bias conditions for this gate. The (0,0) condition does not pose any issues, and the leakage current is that of a HVT transistor. The (1,0) condition is very tricky: the output must stay at Vdd, but is defined by the ratio of the leakage current of a LVT PMOS transistor with $Vsg - |V_T| = 0V - 0.1V = -0.1V$ and the current of a HVT NMOS transistor with $Vgs - |V_T| = 0.5V - 0.45V = 0.05V$. After sufficient time, the output will hence most likely settle to $Vdd/2$. If the activation pulse is short enough, behaviour can still be acceptable. In DSM technologies, the noise margin in this condition will probably not be sufficient. The leakage in this half-selected condition is rather high. The (0,1) condition has a similar leakage problem but has a better noise margin. The full selected condition (1,1) does not pose specific problems.

In [Cos07], we employed a differential medium-swing receiver for the transmission of the block activation signal. As only one signal is transmitted, there are no half-selected conditions. Figure 7.11 shows the circuit schematic. The circuit relies on zero threshold transistors. Their leakage is not an issue as they are either on or biased with negative Vgs. This circuit was used to receive a differential $0.4V$ signal in a $0.18\mu m$ technology in a design with $Vdd = 1.6V$.

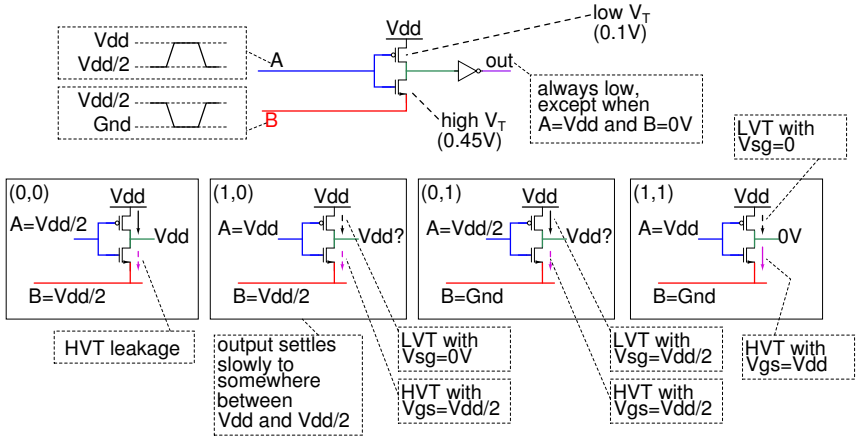


Figure 7.10: Half-swing pulse-mode "AND" gate [Mai98], used as a post-decoder, e.g. as word line driver. Technology node is $0.25\mu m$ and $Vdd = 1V$.

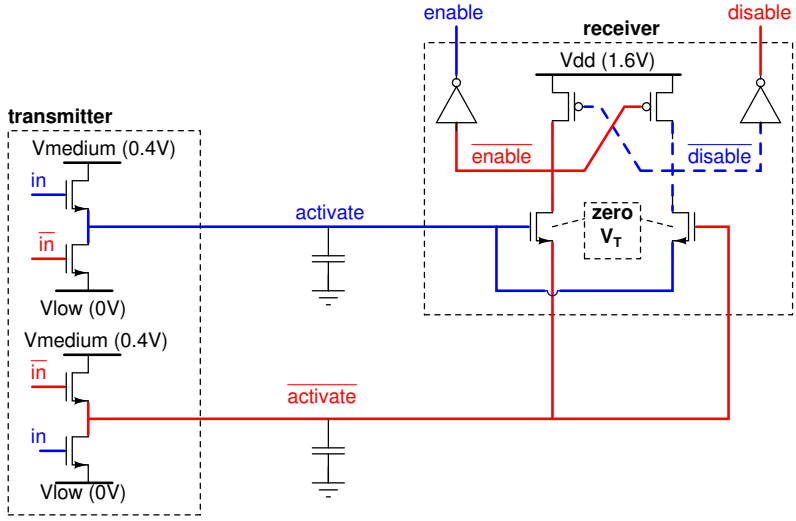


Figure 7.11: A differential medium-swing receiver [Cos07].

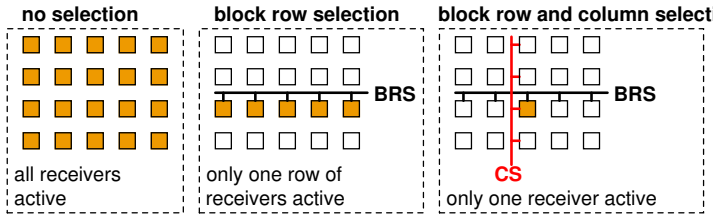


Figure 7.12: When select signals are available prior to the arrival of a low-swing activation signal, fewer receivers have to be active.

To be of use in DSM technologies, a lower swing or a single-ended swing must be supported.

No good fast, self-triggered low-swing receivers have been presented to date. However, such self-triggered receivers that can detect and amplify small input signals (e.g. $200mV$) will require some static current. The fewer receivers are active, the more static current can be tolerated per receiver. Figure 7.12 illustrates that the number of receivers that must be in a sensitive state (and hence require static current) depends on which selection signals are available prior to the transmission of the low-swing activation signal. For example, consider the transmission of a low-swing activation signal to activate a specific local block in a memory which contains 32 block rows and 8 block columns. If no prior selection signal is available, all local blocks must be able to respond to the low-swing activation signal, and all $32 \cdot 8 = 256$ receivers consume static power. If a block row select signal is available beforehand, only the 8 local blocks on that row must be able to respond to the low-swing activation signal, so only 8 receivers consume static power. If a column select signal is available as well, only one receiver consumes static power.

This last situation occurs for example when additional timing events are needed to control the order of operations inside the local block.

7.5 Example implementation

This section outlines the implementation of the decoder and timing that was used in our $90nm$ design [Cos09], a $128Kbit$ memory with 4096 words of 32 bits.

7.5.1 Decoder

Figure 7.13 gives an overview of the three-stage decoder that is used in this design. Three first-level decoders are used to decode the 12 address bits into three sets of one-hot output wires. The *column decoder* (CD) decodes address bits $a_{0:2}$ into 8 column select (CS) wires. The *block row decoder* (BRD) decodes address bits $a_{3:6}$ into 16 block row select (BRS) signals, and the *within block decoder* (WIBD) decodes the remaining address bits $a_{7:11}$ into 32 within block

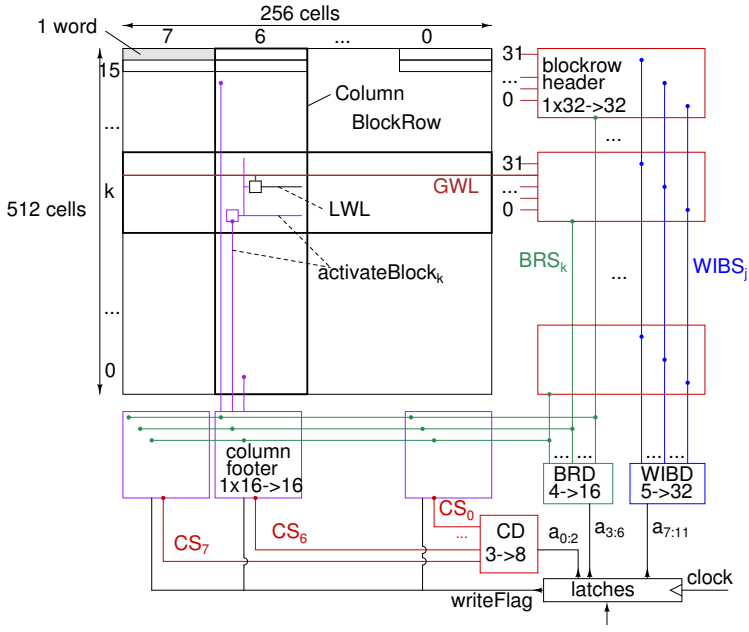


Figure 7.13: Decoder organization for the 128$Kbit$ memory from [Cos09].

signals ($WIBS$). These $WIBS$ signals indicate which word within the selected local block should be enabled.

There are two groups of second-level decoders. In the column footers, the 16 BRS signals are combined with the column select signal CS_i to generate the *activateBlock* signals. Each of these signals is directed to a single local block, where it is buffered before being used to control local block operation. In the block row headers, the 32 $WIBS$ signals are combined with BRS_k to generate the global word line (GWL) signals for this block row. Notice that the BRS signals are used twice in this decoder - once to obtain GWL and once to obtain *activateBlock*. This results in a deeper logic depth for the generation of *activateBlock*, but also in a lower capacitive load and energy consumption. The last level decoder combines the *activateBlock* signal with the GWL signal to activate a single LWL (one out of $2^{12} = 4096 = 8 \cdot 32 \cdot 16$).

Figure 7.14 contains a more detailed view of the decoder. The mentioned capacitances are only the wire parasitics. The length of the wires in some wire sets varies, e.g. for CS_i. In these cases, worst case values are shown here. All decoder stages are AND-type. The decoder consists of static CMOS gates only. Only the small, highly active stages in the decoder and the input latches are implemented low threshold (SPLVT) transistors. This improves the performance at reduced supply voltages without introducing excessively large leakage currents.

The *writeFlag* signal is needed inside the local block. It is routed to the column

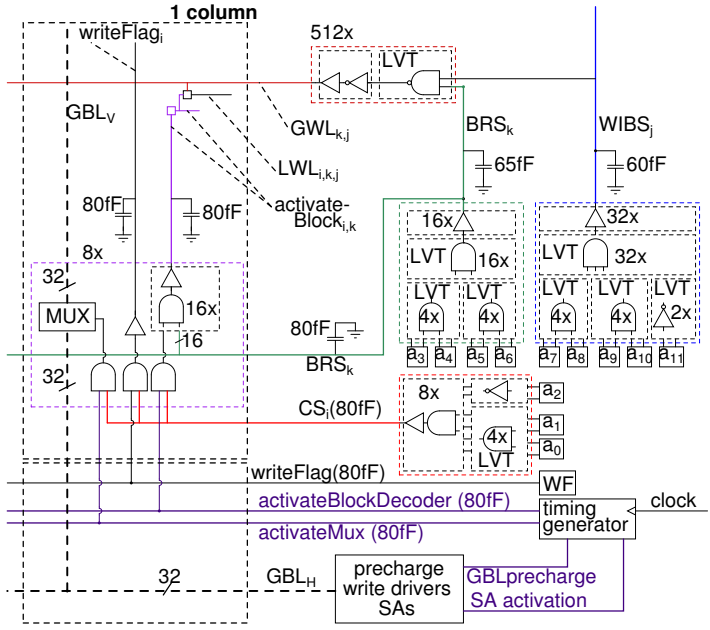


Figure 7.14: Decoder implementation. The multiplier next to each gate indicates the number of instances of this gate in the $128Kbit$ memory from [Cos09].

footer, where it is combined with the static CS signal. The resulting signal $writeflag_i$ is routed to all local blocks in the active column. The CS signal is also used to control the connection between vertical global bit lines GBL_V (one set per column) and the horizontal global bit lines GBL_H (one set for the memory). Timing for this operation is controlled by the $activateMux$ timing signal.

The BRS, $WIBS$ and CS signals only change their values when their input address bits change. However, $activateBlock$ must be a return-zero signal to ensure a correct precharge operation for the local bit lines. This behaviour is obtained by introducing a timing pulse $activateBlockDecoder$ into the decoder gates that generate $activateBlock$ in the column footers. To avoid glitches, the start of this pulse is delayed enough to ensure that it arrives at the local block after both the rising and falling edge of the GWL and $writeFlag_i$ signals.

7.5.2 Operation and timing

Figure 7.15 shows the simulated timing diagram of this memory. All critical timing signals are derived from digitally tunable delay lines. Figure 7.16 shows the implementation of the tunable delay lines used in this design. The delay is programmed using four configuration bits, enabling 16 possible delay values in steps of about $100ps$. The minimal delay value is $100ps$, the maximal delay value is about $1.7ns$. These delay elements are also used in pulse generators to obtain

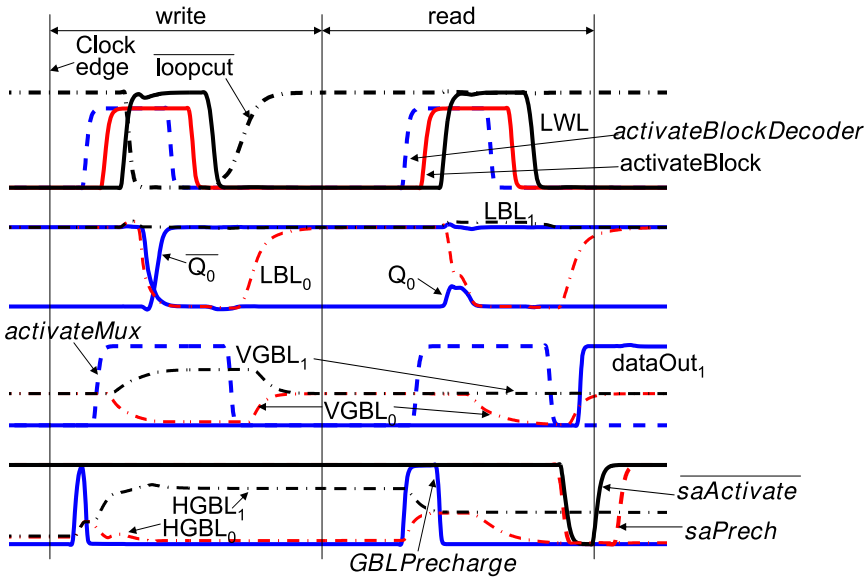


Figure 7.15: Simulated timing diagram.

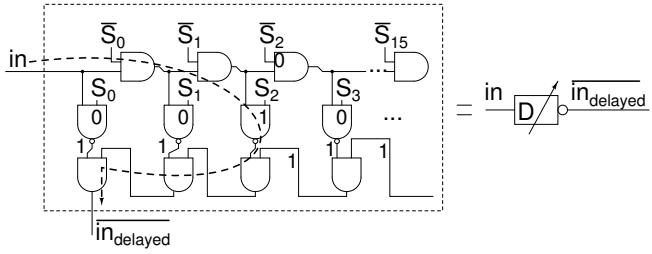


Figure 7.16: Inverting tunable delay line.

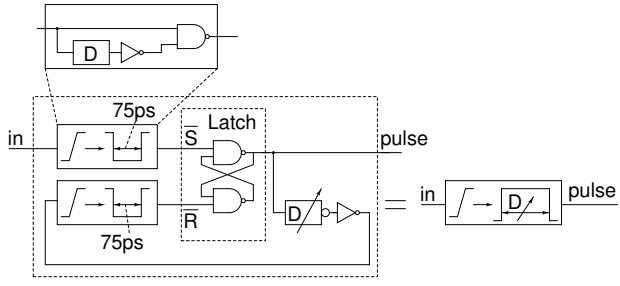


Figure 7.17: Pulse generator with tunable width. This implementation can handle input pulses that are shorter than the desired output pulse width.

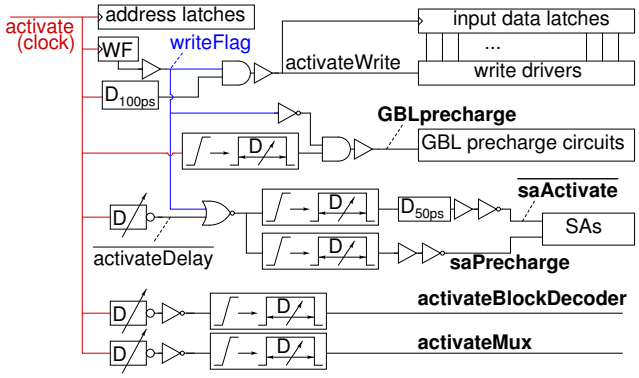


Figure 7.18: Generation of critical timing signals.

a tunable pulse width, as shown in figure 7.17. The current implementation of the tunable delay elements is rather energy hungry. The equivalent capacitance is about $12.5fF$ per $100ps$ segment, which sums up to about $500fJ$ at $1V$ for the total memory timing. This is about 10% of the total energy consumption of this memory. Given their relaxed constraints, it is easy to conceive low power tunable delay elements for future designs. Current starved approaches such as used in [Kim96] appear promising.

Fig. 7.18 shows how the different critical timing signals for this memory are generated. Only the rising edge of the clock signal is used. In this design, eight fully tunable delay lines are used to generate five important control signals, displayed in bold in the figure. This large number of tunable delay elements provides maximum timing flexibility and reduces the layout effort. The price paid for this flexibility is energy: the tunable delay system consumes 10% of the total memory energy consumption during read accesses. Future designs can reduce this contribution significantly, either by using better delay elements or by removing the flexibility that is not strictly needed.

7.6 Summary

This chapter covers the decoder and the timing control.

Section 7.2 discusses the decoder, the circuit that decodes the address bits and selects the corresponding block and word. Different organisations for the global decoder are compared and the implementation of the predecoders is illustrated.

Section 7.3 treats the timing control. During a memory access, several timing events occur, such as the activation of the word line and the activation of the global bit line precharge. To ensure correct operation, these events must satisfy several delay constraints. Typical timing events and their delay constraints for a low power memory are provided.

Different approaches can be used to ensure the correct order of these events.

The available options are discussed. First, timed delays can be generated by local timing circuits, or they can be generated by global timing circuits, in which case they still need to be transmitted to the correct local block. Second, more or fewer external clock events can be used as inputs for the timing control circuits. Third, three methods can be used to ensure that the correct operation order between events is respected: the simulator-based approach, the replica-based approach and the calibration-based approach. Calibration-based timing allows to use optimal delay settings for the actual die and operation mode without introducing design-time risks.

A large part of the energy for the decoder and timing is associated with the transmission of signals across the memory matrix. Section 7.4 presents schemes for the reduced-swing transmission of activation and selection signals. To enable the use of reduced-swing transmission of activation signals in DSM technologies, better receiver implementations are required.

Section 7.5 discusses the integrated timing and decoder solution that is used in our second prototype [Cos09].

Contributions and highlights

This work introduces calibrated timing as a solution for the increasing delay uncertainty due to transistor variations. Calibrated timing also enables optimal timing settings for different operation modes.

Chapter 8

Prototypes

8.1 Introduction

Two prototype memories have been successfully designed, fabricated and measured to validate the assumptions and findings provided in this work. Section 8.2 outlines the first prototype, section 8.3 discusses the second prototype. Section 8.4 compares the performance of both prototypes with the current state of the art.

8.2 First prototype ($0.18\mu m$)

A self-contained description of this prototype memory is provided in [Cos07]. The relevant highlights and measurement results are outlined here.

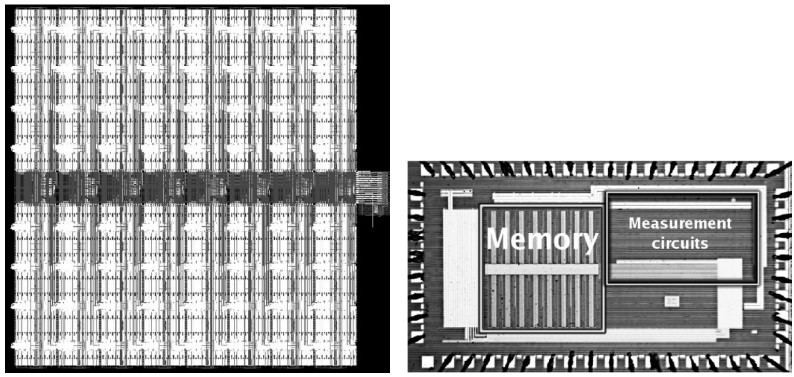
8.2.1 Design description

Table 8.1 summarises the memory dimensions. The main power supply voltage for the design is $Vdd = 1.6V$. Figure 8.1 shows the layout and chip photograph. This section outlines the techniques that are used in this design.

5T SRAM cell using dynamic read stability. The $4Kword$ memory with 16 bits per word employs a single-ended $5T$ SRAM cell (figure 2.59), which relies on dynamic read stability (section 3.6) for its correct operation. For a read operation, the short local bit line is precharged to $Vdd_{LBL} = 1V$. The local bit line is discharged before the cell suffers from a read upset. To write

technology	$0.18\mu m$; $1P6M$
threshold voltages	regular and zero V_T
word length	16
memory size	4Kword ; 64Kbit
cell type	5T single-ended SRAM cell
cell size (logic DRC rules)	$6.06\mu m^2$ [$187F^2$]
memory core size	$690000\mu m^2$
effective cell size	$10.5\mu m^2$ [$325F^2$]

Table 8.1: Memory dimensions.



(a) Layout. (b) Chip photograph.

Figure 8.1: The fabricated $64Kbit$ memory chip. The memory measures $850\mu m$ by $810\mu m$.

a "1" to the cell, the local bit line is forced to Vdd_{LBL} during the entire cell access, which results in a correct write operation.

Short buffered local bit lines with low-swing extended GBL. A short single-ended local bit line with readbuffer is used, as discussed in section 3.5.6. Each local bit line connects to 16 cells. $VGBL$ and $HGBL$ are precharged at a low voltage ($0.4V$) and are connected with pass transistors. The readbuffer discharges both. This setup for the read data transfer is discussed in section 6.2.2.3.

Low-swing write using shared low-swing receivers. A Low-swing signal on the GBL during write is made possible by the use of a low-swing receiver (section 6.3.1). The low-swing receivers are shared between four local bit lines, as discussed in section 3.5.6. The low-swing receivers are gate-input stacked latch-type SAs (section 4.6.2).

Self-closing sense amplifiers. The global read sense amplifiers are self-closing SAs (section 4.6.4).

Distributed decoder. A distributed decoder with reduced swing signalling is used. As discussed in section 7.4, the activation signals are transmitted as a differential medium-swing signal ($0.4V$). These signals are detected by a receiver which is triggered by this activation signal. The signals which only carry binary information, not timing information, are received by self-closing SAs which are triggered by the activation signal. A single-ended $0.4V$ signal is used.

Figure 8.2 provides an example of a low-swing distributed decoder for a $8Kword$ memory with 32 bit word length. In this example, the $256Kbit$ matrix is divided in 8 by 8 blocks, each containing 128 words of 32 bits. In the input block of the memory, the address bits used to select the block are decoded. In this design, it is assumed that the highest address bits a_{11},a_{12} and a_{13} have a low activity, as

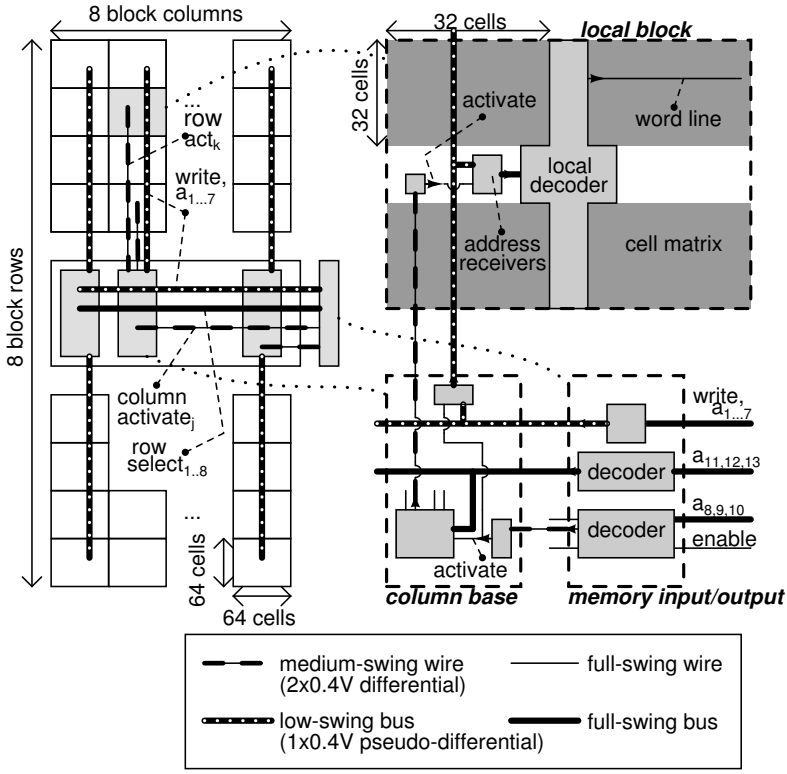


Figure 8.2: A low-swing distributed decoder for a $256Kbit$ memory with 32 bit word length ($8Kword$).

is the case for applications with good locality of reference. Therefore, they are statically decoded into the row select signal which is transmitted in full swing to all column bases. Because of the low activity, this has only a very limited impact on the total energy consumption. In parallel, address bits a_8, a_9 and a_{10} are decoded into the column select signal. This signal is combined with the activate signal and is then sent to the base of the appropriate column using a medium swing transmitter and receiver. Also in parallel, the remaining address bits a_1 to a_7 and the write enable signal *write* are put on the horizontal low-swing address bus, which is at this time isolated from all vertical address busses by means of pass transistors. Next, the column activate signal is received at the base of the column. The pass transistors between the horizontal address bus and the appropriate vertical address bus are enabled. Based on the row select that is already available, a medium swing block select signal is sent to the appropriate block. By the time the block select signal is restored to a full level signal by the medium swing receiver in the local block, the data on the vertical address bus is available. The activate signal enables the low-swing receivers, which are implemented as sense amplifiers. The address bits a_1 to a_7 are then decoded within the local block. All needed control signals, such as the precharge signals for the local bit lines, are generated locally. The area overhead of this scheme is limited. The local decoder is repeated in each local block and eight sense amplifiers are added to each block.

The decoder in the actual prototype is organised as 64 blocks (8 rows by 8 columns) of 32 cells by 32 cells.

8.2.2 Measurement results

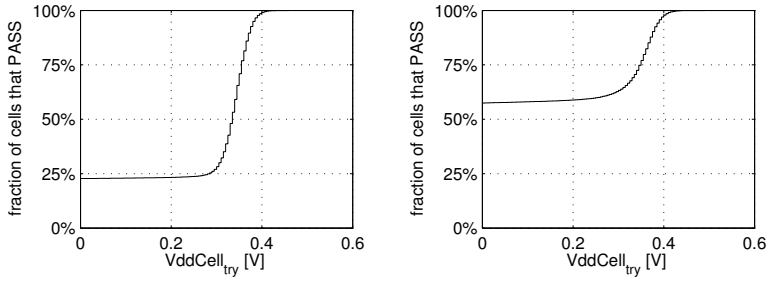
Cell measurements: functional operation under intra-die variations

To verify the functional operation of the cells under intra-die variations, four measurements are performed on all cells of a die. The first two measurements assess the data retention voltage for the cells. The third measurement assesses the available margin against destructive reads. The fourth measurement assesses the available write margin.

For the first measurement, a zero is written to all cells using a known good supply voltage for the cell. Next, the cell supply voltage is temporarily reduced to $VddCell_{try}$. Finally, the cell contents is read out using a known good supply voltage for the cell. This is repeated for decreasing values of $VddCell_{try}$. Figure 8.3a shows the measurement results for all 65536 cells on a single die.

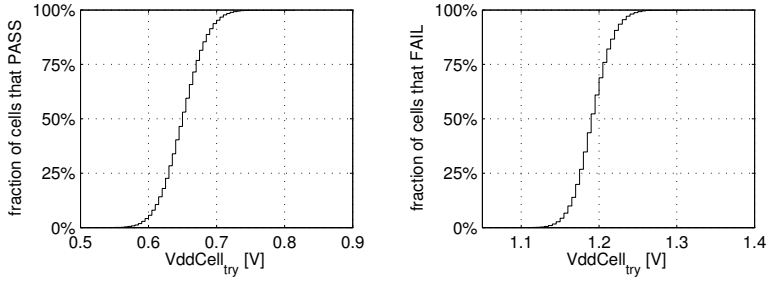
The second measurement is identical to the first measurement, but now a high value is written to the cells. Figure 8.3b shows the measurement results for all cells on a single die. A relatively large amount of cells have a high value as preferential state, which could have been expected given the asymmetric cell sizing and the high voltage on the local bit line.

These first two measurements indicate that these cells should not be operated below a cell supply voltage of $0.5V$. Since this design never uses such low supply



(a) original state "0". (b) original state "1".

Figure 8.3: Measured fraction of cells that correctly retain their original state when the cell supply voltage is temporarily reduced. ($V_{BL} = 1.2V$, $V_{WL} = 0V$)



(a) Read "0". (b) Write "1".

Figure 8.4: (a) Measured fraction of cells that do not suffer from destructive read when operated at a reduced cell supply voltage ($V_{BL,precharge} = 1.2V$, $V_{WL} = 1.6V$). (b) Measured fraction of cells that are not written correctly at a given cell supply voltage ($V_{BL,write} = 1.2V$, $V_{WL} = 1.6V$).

voltages for the cells, this does not pose any problems.

The third measurement identifies for each cell the highest cell supply voltage at which destructive read occurs. For this measurement, a zero is written to all cells using safe supply voltages. The cell supply voltage is reduced to $VddCell_{try}$ and a read access is performed using this reduced cell supply voltage. This is repeated for decreasing values of $VddCell_{try}$. Figure 8.4a shows the measurement results for all cells on a single die. The distribution of $VddCell_{try}$ at which cells fail has a mean of $650mV$ and a spread of $32mV$. These measurements show that cell supplies larger than $850mV$ (6-sigma) do not result in destructive reads.

The fourth measurement identifies for each cell the lowest cell supply voltage at which it can no longer be written correctly. A zero is written to all cells. The cell supply voltage is increased to $VddCell_{try}$, and a write access is performed, attempting to write a high value to the cell. Afterwards, the cell state is exam-

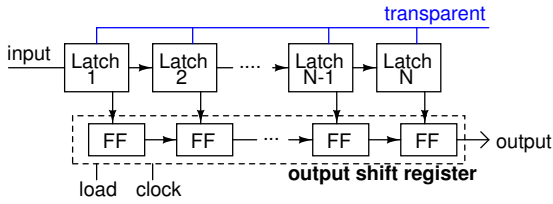


Figure 8.5: An on-chip digital oscilloscope.

ined using safe voltages. This is repeated for increasing values of $VddCell_{try}$. Figure 8.4b shows the measurement results for all cells on a single die. The distribution of $VddCell_{try}$ at which cells are no longer writeable has a mean of $1.192V$ and a spread of $23mV$. These measurement show that cell supplies smaller than $1.050V$ (6-sigma) allow all cells to be written. These results are in sufficient accordance with the simulations. The results from the last two measurements show that in this design, the application of dynamic read stability on a cell which does not have any SNM_{read}, creates sufficient stability margin.

Performance measurements

The available number of bondpads and the measurement equipment does not allow to perform continuous at-speed measurements of the memory. Rather, the write data and address bits are first shifted serially into an input shift register. A single access is performed, and the results of this access are stored in output shift registers. These output shift registers are than serially read and interpreted by a software program.

The memory access time is measured with an on-chip digital oscilloscope as shown in figure 8.5. The signals that are to be measured are applied at the inputs of a latch line. Before the memory is activated, the latch line is made transparent by asserting *transparent*. When the input signal changes, this change ripples through the latch line with a delay T_{latch} per latch element. After the memory access is completed, the latch line is frozen by lowering *transparent*. The nth latch contains the value of the input signal at a time $n \cdot T_{latch}$ before the latch line was frozen. T_{latch} must be obtained from another measurement. In this design, this value is obtained by sampling a known external clock with an identical latch line.

Both the SA output and the memory activation signal are monitored with such a latch line. The memory access time can then be assessed as the difference between the memory activation time and the time at which the SA output changes. Other latch lines are added to monitor interesting internal signals, such as a specific word line and internal activation signals.

The state of the latch lines is loaded into an output shift register and interpreted by a software program.

With such a measurement approach, the maximal operation frequency can not

technology	$0.18\mu m$
word length	16
memory size	4Kword ; 64Kbit
supply voltages[1]**.**	
$V_{local\ bit\ line}$	$1.2V$
$Vdd_{decoder} = Vdd_{WL} = Vdd_{SA}$	$1.6V$
$V_{prech,GBL}$	$0.4V$
$V_{ref,SA}$	$0.275V$
access time	$4ns$
energy per access	$9.5pJ$

Table 8.2: Measured performance values for the first prototype memory.

be measured directly, but must be derived from the access time measurement. A disadvantage of this approach is that problems with the precharge between two accesses cannot be detected. Simulations show that there are no such problems, but direct measurements are much more preferable.

Table 8.2 shows the measurement results for the fabricated $64Kbit$ memory. The access time is $4ns$ and the maximal operation frequency is $250MHz$. If all charge is derived from a single $1.6V$ power supply with ideal series regulators, the energy consumption is $9.5pJ$ per access ($2.4mW$ at $250MHz$). If the charge for the global bit lines is taken from a dedicated power supply at $0.4V$, only $6.8pJ$ per access is consumed ($1.7mW$ at $250MHz$). For the energy consumption, a random data pattern with $p_0 = p_1 = 50\%$ and a row select signal activity of $1/16$ is assumed.

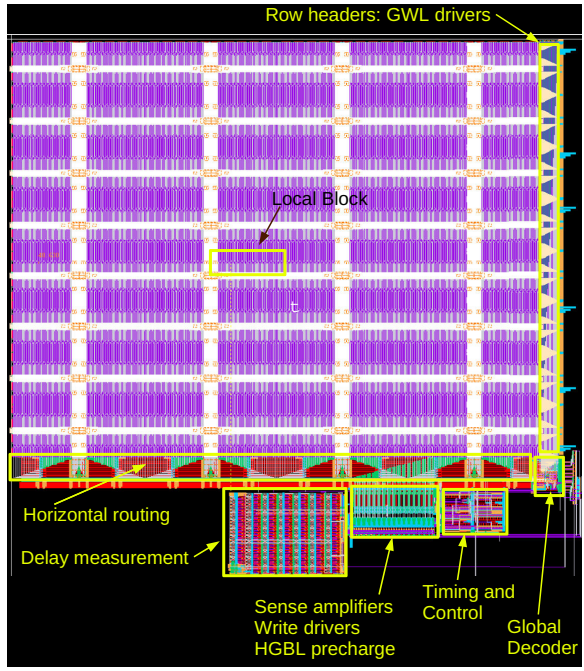
8.3 Second prototype ($90nm$)

A self-contained description of this prototype can be found in [Cos09]. The relevant highlights and measurement results are outlined here.

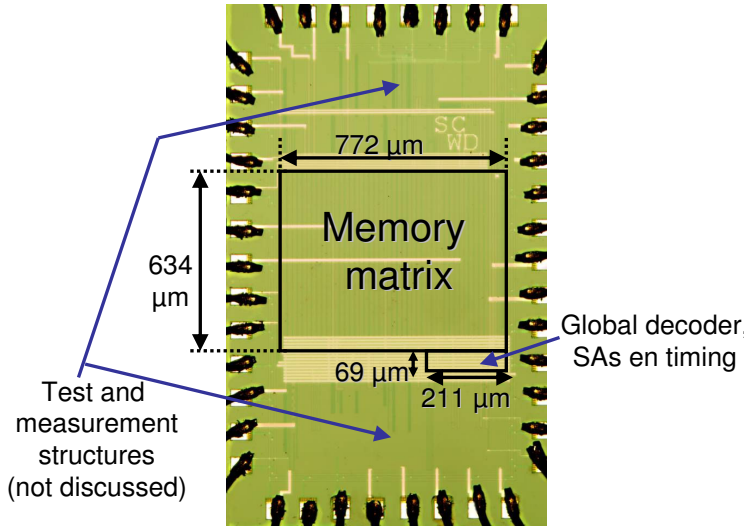
8.3.1 Design description

Table 8.3 summarises the memory dimensions. Figure 8.6 shows the chip photograph. This section outlines the techniques that are used in this $4Kword$ $128Kbit$ design.

General organization. Figure 8.7 shows the global memory organization. The memory matrix is build up of local blocks, which each contain 32 words (1K cells). The top level matrix is subdivided in 8 columns by 16 rows of these local blocks. For each two local blocks there is a shared write block, which contains the write receivers. These receivers enable low swing signals on the global bit lines during write operations.



(a) Layout.



(b) Chip photograph.

Figure 8.6: The fabricated 128$Kbit$ memory chip.

technology	$90nm$; $1P9M$; multi V_T
word length	32
memory size	4Kword ; 128Kbit
cell type	single-ended SRAM cell pull-down path blocked during write
cell size (logic DRC rules)	$1.51\mu m$ x $1.3\mu m = 1.96\mu m^2$ [$242F^2$]
memory matrix size	$772\mu m$ x $634\mu m = 490000\mu m^2$
total memory size	$504000\mu m^2$
effective cell size	$3.85\mu m^2$ [$475F^2$]

Table 8.3: Memory dimensions for the second prototype.

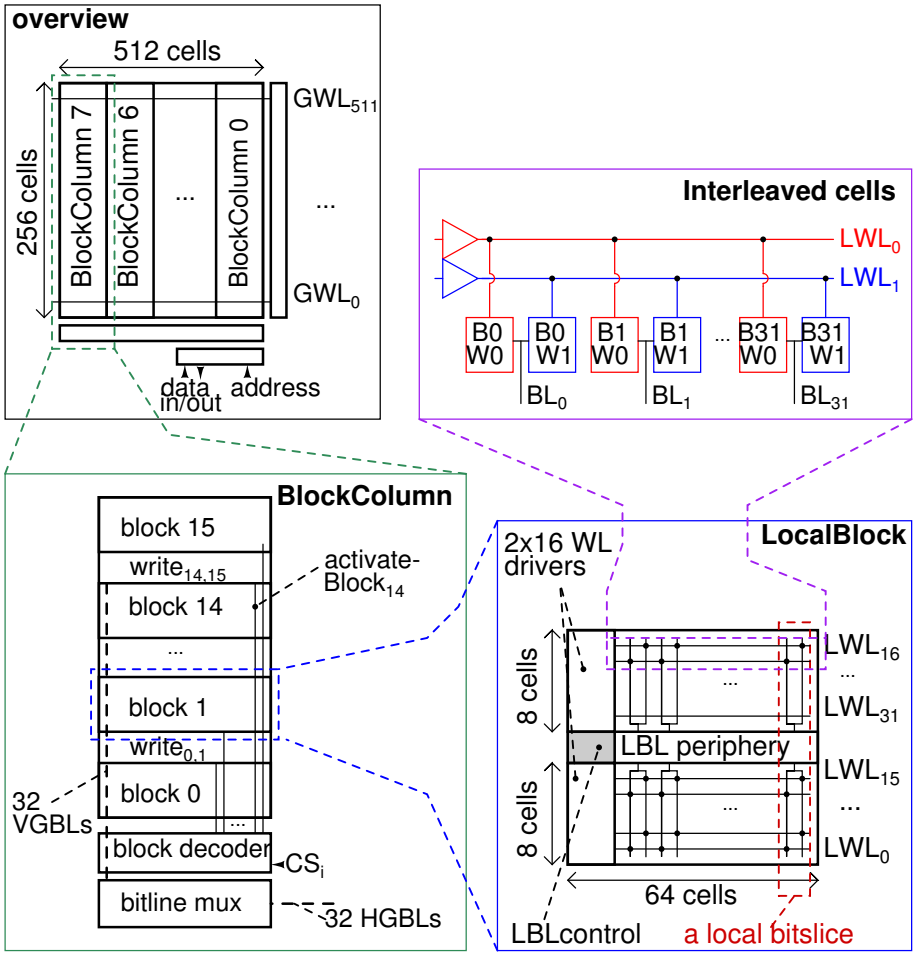
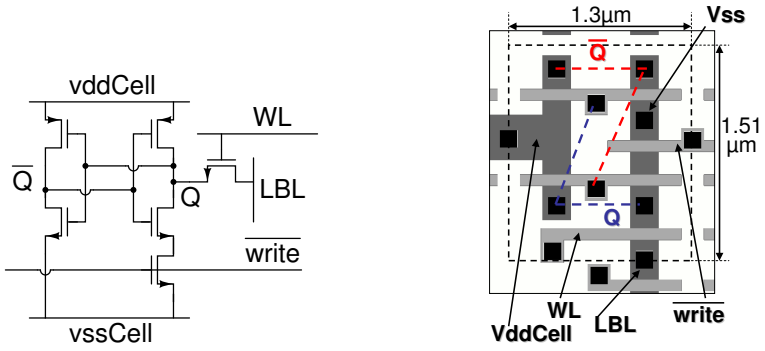


Figure 8.7: Global memory organization.



(a) Cell schematic.

(b) Cell layout.

Figure 8.8: Single-ended cell with write assist transistor that cuts the pull-down path during write. All transistors are SPHVT and have $W = 200nm$ and $L = 80nm$.

The memory is based on a fully subdivided WL architecture, so each word has its own LWL. This avoids the problems associated with the half-select condition for cells and ensures that only the required cells and bit lines are activated, which is crucial to obtain a low energy per operation.

In figure 8.7, there are 512 GWLs for 256 cell rows. This apparent inconsistency is caused by the fact that the cells of two words are interleaved as shown in figure 8.7 . Each WL spans 64 cells, but is only connected to 32 of them. This requires two WLs per cell row. This approach reduces the bit line capacitance: for a given number of cells connected to the bit line, the bit line wire length is halved, which reduces its parasitic capacitance.

Decoder. The decoder that is used in this design is detailed in section 7.5.

Bit line hierarchy. The design employs a short buffered bit line with low-swing extended global bit line as discussed in sections 3.5.7, 6.2.2.3 and 6.2.3. Shared low-swing write receivers are employed.

Operation and timing. The operation and the timing control for this design are detailed in section 7.5.

Tunable global read sense amplifiers. The tunable sense amplifiers that are used in this design are discussed in section 5.4.2. A pulsed current source SA (section 4.6.5) is used in combination with a tuning system with 16 reference voltage levels. The calibration phase is controlled off-chip.

Single-ended cell. The single-ended cell from figure 8.8. A write assist transistor cuts the pull-down path during write operations.

Selective voltage scaling. Reducing the supply voltage reduces both dynamic energy consumption and leakage power. Therefore, it makes a lot of sense to reduce the supply voltage when delay slack is available. However, reducing the

supply voltage of all circuits at once is not the best solution if one aims at the lowest possible energy consumption at reasonable speeds.

In this design, Vdd_{cell} and Vdd_{WL} combined consume less than 5% of the total dynamic energy consumption (see table 8.5). Reducing these two supplies would only have a limited effect on the dynamic energy consumption (less than 5%). However, as the cells are implemented with the highest available threshold voltage to reduce leakage in the most effective way, reducing these supplies (during operation) would be detrimental for the memory speed. Below a certain voltage, cell functionality is endangered as well. The other circuits, such as the decoder and the SAs are typically implemented with lower threshold voltages, with larger transistors and without use of ratioed logic. Reducing only the supply voltages for these circuits, not for the memory cells, results in a smaller delay penalty and less robustness issues. Because of this reduced impact, lower supply voltages can be used. From this example, it is clear that the best results are obtained when one is selective in which voltages to reduce. Selective voltage scaling has been applied before in memories, most often to avoid stability and reliability issues and always under different names. In [Pil08] for example, the technique is labeled as a dual supply SRAM array.

Notice that the arguments against a reduced cell voltage apply only to active cells. Cells that are not accessed have significantly larger margins, and reducing their cell supply can be a very effective way to reduce leakage power [Gee08].

This second prototype uses separate supplies for cells and WLs. These voltages are reduced much less aggressively than the main power supply. A standby mode was not implemented.

8.3.2 Measurement results

Sense amplifier measurements

The standard deviation of the measured offset voltage of the 32 SAs on a single die is $\sigma_{V_{offset}} = 19mV$. As mentioned in table 5.2, SA calibration allows to operate the memory with a single-ended global bit line swing of $40mV$ when a relaxed memory latency is used.

Cell measurements

Figure 8.9 indicates that the cells operate reliably under intra-die variations, although the limited write margins suggest the use of a double-ended write for designs in future technology nodes. Such a change would have little impact on the performance.

Performance measurements

Table 8.4 shows the measured speed and energy consumption of the memory in the different operation modes. The maximal operation speed is derived from the measurement of the memory access time, as explained for the first prototype.

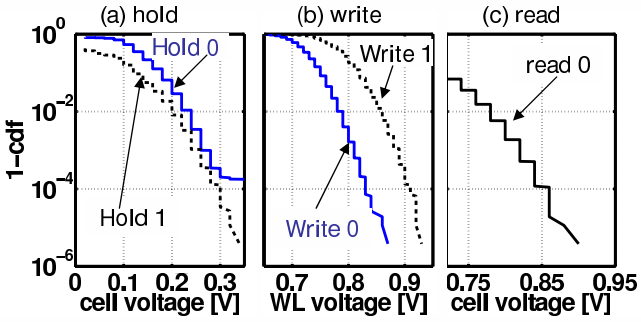


Figure 8.9: Measured cumulative distribution of cell operation margins for all 128K cells on one die. (a) Fraction of cells that fail to hold data at reduced Vdd_{cell} [$V_{WL} = 0V$]. (b) Fraction of cells that can not be written correctly at reduced V_{WL} [$Vdd_{cell} = 0.9V$, $V_{LBL} = 0.6V$]. (c) Fraction of cells that are not read correctly at reduced Vdd_{cell} [$V_{WL} = 1.2V$, $V_{LBL} = 1.0V$].

main Vdd	[V]	1.2	1.0	0.8	0.6
maximum frequency	[MHz]	850	780	480	240
static energy per access	[pJ]	1.3	0.8	0.5	0.7
active energy per access	[pJ]	7.1	4.5	3.1	2.0
total energy per access	[pJ]	8.4	5.3	3.6	2.7
leakage power	[μW]	1123	625	219	162

Table 8.4: Measured performance values for the second prototype memory. The static energy per access is calculated by integrating the leakage power over one idle clock period at the maximum clock frequency.

Measurement results [1]																
Maximum frequency [MHz]	850				780				480				240			
Main/Wordline/Cell Vdd [V]	1.2 / 1.2 / 1.2				1.0 / 1.2 /1.0				0.8 / 1.0 / 0.9				0.6 / 1.0 / 0.9			
	read		**write**		**read**		**write**		**read**		**write**		**read**		**write**	
active energy / access [fJ]	6200	100%	8935	100%	3880	100%	5833	100%	2815	100%	3686	100%	1965	100%	2074	100%
Global decoder + GBL control	1140	18%	2160	24%	610	16%	1105	19%	376	13%	684	19%	204	10%	369	18%
Sense amplifiers	869	14%	0	0%	583	15%	0	0%	403	14%	0	0%	309	16%	0	0%
cells	84	1%	208	2%	0	0%	153	3%	0	0%	75	2%	0	0%	41	2%
WL and write WL (writeBar)	84	1%	150	2%	84	2%	150	3%	70	2%	125	3%	70	4%	125	6%
local bit lines, readbuffer, write bit lines and receivers	750	12%	1638	18%	415	11%	1163	20%	262	9%	614	17%	165	8%	254	12%
global bitlines (100mV swing)	516	8%	1176	13%	340	9%	990	17%	264	9%	772	21%	381	19%	522	25%
local control and decoder	879	14%	2373	27%	583	15%	1438	25%	368	13%	776	21%	197	10%	416	20%
timing (includes delay elements and control wires)[2]	1878	30%	1230	14%	1265	33%	835	14%	1072	38%	640	17%	639	33%	348	17%
	static				**static**				**static**				**static**			
static energy / access [fJ] [3]	1322	100%			801	100%			457	100%			674	100%		
Global decoder + GBL control [4]	236	18%			82	10%			-16	-3%			-145	-22%		
Sense amplifiers	4	0%			4	1%			2	0%			2	0%		
cells	330	25%			96	12%			102	22%			205	30%		
WL and write WL (writeBar)	414	31%			454	57%			255	56%			511	76%		
local bit lines, readbuffer, write bit lines and receivers	76	6%			45	6%			38	8%			37	5%		
global bitlines	3	0%			4	0%			5	1%			8	1%		
local control and decoder	254	19%			114	14%			69	15%			55	8%		
timing (includes delay elements and control wires)	5	0%			2	0%			2	0%			2	0%		
	read		**write**		**read**		**write**		**read**		**write**		**read**		**write**	
Total energy/access [fJ]	7522		10257		4681		6634		3272		4143		2638		2748	
	average				**average**				**average**				**average**			
average[5] energy/access [fJ]	8433				5332				3562				2675			
leakage power [uW][6]	1123				625				219				162			

[1] At room temperature. All energy values calculated as Q*VddMain, except for WL and cells.

[2] includes the tunable delay elements (~500fF during write, 550fF during read), but also the associated control wires and controlled transistor gates: activateBlockDecoder (125fF), activateMux(125fF), GBLPrecharge(100fF), saActivate (150fF) and saPrecharge (200fF). Capacitances were estimated by integrating the currents in back-annotated simulations

[3] The static energy per access is calculated by integrating the leakage power over one clock period at the maximum frequency

[4] There is significant leakage from the WL supply to the global decoder supply

[5] 1/3 write, 2/3 read

[6] this leakage power is already included in total and average energy/access

Table 8.5: Detailed measured energy consumption for the second prototype.

In this test setup, T_{latch} is obtained by constructing a ring oscillator with an identical copy of the latch line as delay element. Off-chip measurement of the frequency of the ring oscillator allows to calculate T_{latch}, which is around $100ps$.

Table 8.5 gives a more detailed picture of the energy consumption. Even though the WL and cell voltage are not scaled down, their energy consumption remains small. The limited currents make it feasible to generate these supplies with an on-chip converter in a future design. The memory covers a wide performance range, consuming between 8.4pJ/access at 850MHz and 2.7pJ/access at 240MHz.

8.4 Comparison with the State of the Art

Table 8.6 compares the performance of the prototypes that were developed in this work ([Cos07, Cos09]) with that of other state-of-the-art memories. This comparison is necessarily limited due to the fact that many publications on memories fail to provide adequate information about their dynamic energy consumption and matrix organisation. For each of the memories, the original published data is tabulated first. For ease of comparison, specifications for a $90nm$ implementation have been estimated. To convert from technology node F with supply voltage Vdd_A to the $90nm$ technology node, we first estimated the most likely supply voltage for a corresponding $90nm$ implementation (Vdd_{90nm}). The energy consumption was then estimated as

$$E_{90nm} = E_F \cdot \frac{C_{90nm}}{C_F} \cdot \left(\frac{Vdd_{90nm}}{Vdd_F}\right)^2 = E_F \cdot \frac{90nm}{F} \cdot \left(\frac{Vdd_{90nm}}{Vdd_F}\right)^2$$

This scaling rule is too optimistic when scaling from older technology nodes to $90nm$, as it assumes that bit line swings and other low-swing signalling can be reduced together with Vdd, while it actually becomes harder to sense even a fixed low-swing voltage. Furthermore, increasing intra-die variations complicate timing and bit line swing control.

To simplify the comparison between memories of different size and word length, two figures of merit are provided in the table: $\frac{\mathrm{E}}{\mathrm{WL}}$ and $\frac{\mathrm{E}}{\mathrm{WL}\sqrt{\mathrm{size}}}$. The first figure compensates for the fact that accessing more bits requires more energy. However, as the decoder, timing and leakage add an energy overhead that does not change significantly with WL, this metric tends to be more favorable for memories with wide WL. The second figure also compensates for the fact that all capacitances become larger in larger memories. For the most relevant capacitances, the increase is proportional to the height or the width of the memory array. In a square implementation, this is $\sqrt{size}$. As long as leakage is not dominant, this metric tends to favor larger memories, as it is more rewarding to use more advanced low-swing schemes in such memories.

Our second prototype ([Cos09]) is a $128Kbit$ memory with $WL = 32$ in $90nm$ technology that supports different operation modes. In the $1.2V$ mode, the maximum frequency is $850MHz$, $\frac{E}{WL} = 263\frac{fJ}{bit}$ and $\frac{E}{WL\sqrt{size}} = 725\frac{aJ}{bit^{1.5}}$. In the

design	node $[nm]$	size $[Kbit]$	WL $[bits]$	Vdd $[V]$	freq $[MHz]$	E $[pJ]$	$\frac{E}{WL}$ $[\frac{fJ}{bit}]$	$\frac{E}{WL\sqrt{size}}$ $[\frac{aJ}{bit^{1.5}}]$
[Cos07]	180	64	16	1.6	250	9.5	594	2319
(this work)	**90**			**1**	**500**	**1.9**	**116**	**453**
[Cos09]	**90**	128	32	**1.2**	**850**	**8.4**	**263**	**725**
(this work)	**90**			**1.0**	**780**	**5.3**	**166**	**457**
	90			**0.8**	**480**	**3.6**	**113**	**311**
	90			**0.6**	**240**	**2.7**	**84**	**233**
[Tak06]	**90**	64	16	**1.0**	**833**	**12.9**	**806**	**3149**
	90			**0.7**	**500**	**6.3**	**394**	**1538**
	90			**0.5**	**50**	**3.2**	**200**	**781**
[Yan05]	250	256	32	2.5	220	135	4219	8240
	90			**1.0**	**611**	**7.78**	**243**	**475**
[Ver08b]	65	256	128	0.35	0.03	88	688	1342
	65			0.5	0.2	30	234	458
[Sin08a]	65	64	128	1.2	200	23	180	702
	65			0.4	0.5	10.5	82	320
	90			**0.4**		**13.8**	**108**	**420**
[Mai98]	250	32	16	2.4	N/A	55	3438	19000
	250			1	100	9	563	3107
	90			**1**	N/A	**3.44**	**215**	**1187**
(see text)	**90**			**0.5**	**?**	**0.81**	**51**	**280**
[Bha03]	130	256	64	1.5	1000	87.1	1361	2658
	130			0.9	500	26.9	420	821
	90			**1**	**1440**	**26.8**	**419**	**817**
	90			**0.7**	**720**	**11.3**	**176**	**344**
[Cal07]	65	256	128	0.4	0.46	7.18	56	110
[Sha09]	130	40	20	0.9	100	6.5	325	1605
	90			**0.7**	**100**	**2.7**	**136**	**673**
[Kus09]	65	256	256	0.9	127	35	137	267
	65			0.7	36	22	86	168
	90			**0.9**	**100**	**48**	**189**	**370**
	90			**0.7**	**30**	**30**	**119**	**232**
[Ram09]	45	512	16	1	2200	31.8	2000	2745
	90			**1**		**42**	**2625**	**3625**

Table 8.6: Comparison with other state-of-the-art designs. For each design, the original published data is tabulated first. For ease of comparison, specifications for a $90nm$ implementation have been estimated. WL is the word length, $freq$ is the maximal frequency for the specified operation mode, E is the total energy consumption (static+dynamic) per access at the specified clock frequency.

$0.6V$ mode,the maximum frequency is $240MHz$, $\frac{E}{WL} = 84\frac{fJ}{bit}$ and $\frac{E}{WL\sqrt{size}} = 233\frac{aJ}{bit^{1.5}}$. Selective voltage scaling and calibration-based timing enable an efficient run-time trade-off between energy and delay.

The $64Kbit$ memory with $WL = 16$ in $90nm$ that was discussed in [Tak06] consumes more energy at equal speed than our second prototype, even though it is smaller and has a smaller WL. The performance after scaling of [Yan05] is close to that of our second prototype, but as mentioned before, the assumption of ideal scaling from $0.25\mu m$ to $90nm$ is too optimistic.

The memory in [Ver08b] is designed to operate in sub-threshold, while the memory in [Sin08a], from the same group, operates from sub-threshold voltages up to $1.2V$. For this second memory, the lowest energy per operation occurs at $0.4V$. Even though $WL = 128$, the energy per bit is not lower than that of our prototype.

In [Mai98], half-swing pulse-mode techniques are used in the decoder and during write operations also in the bit lines. This requires a second supply at $\frac{Vdd}{2}$ which does not have to deliver large currents though. The design also required negative threshold transistors. It is unlikely that the design would actually scale gracefully to $90nm$, $Vdd = 0.5V$. Still, it is a design with some very interesting features.

The design in [Bha03] is implemented in a $130nm$ SOI technology. At reduced supply voltage, it provides a reasonable balance between performance and energy consumption. Notice that this design benefits from the reduced drain capacitance in the SOI technology, and is favored by the metrics due to its larger WL of 64 and due to its larger matrix size. The sub-threshold memory in [Cal07] has a significantly lower energy per bit than our prototype, but its operating frequency is only $460KHz$ and there are some unresolved reliability issues. A large part of the reduction in energy per bit is also due to the large WL of 128. Because the energy consumption is mainly caused by leakage, it is virtually impossible to provide an honest conversion to $90nm$.

The energy consumption of [Sha09] is good, but even when compensating for the smaller WL of 20, its energy consumption per bit is not lower than that of the low-power mode of our prototype, which is 2.4 times faster even though our matrix is larger ($128Kbit$ versus $40Kbit$).

The energy metrics of [Kus09] are good, in part due to a wider WL. This memory is significantly slower than our second prototype though. The memory in [Ram09] is added as an example of the type of trade-off that is made in more high-performance memories.

From this comparison, it is clear that our second prototype outperforms other state-of-the-art memories when a low energy consumption per operation is required at access speeds between $100MHz$ and $850MHz$.

Chapter 9

Conclusions

9.1 Summary of the chapters

This text discusses all the important parts of low-power SRAM memories at circuit level and attempts to highlight the important interaction between these different parts. The major new challenges that are introduced by deep sub-micron technologies are explicitly addressed: large uncorrelated intra-die variations, leakage currents and wire dominance. Reliability issues due to time-varying effects are considered to a lesser extent. Analysis techniques to assess the impact of variations are outlined, and variability-resilient circuits and techniques receive special attention.

Chapter 2 addresses the core building block of the SRAM memory, the **memory cell**. The important cell specifications are outlined, the traditional analysis methods are presented and an overview of methods to analyse the behaviour of SRAM cells under transistor mismatch is provided. The design options for the traditional 6T SRAM cell are explored, with special attention to the impact of bias voltages on cell stability and write-ability. A brief introduction to SRAM cell layout and local wire routing is included as well. The chapter concludes with an overview of interesting alternative SRAM cell types.

Two alternative SRAM cell types have very interesting properties: the 8T cell with readbuffer is interesting as it circumvents all read stability issues, while cells with single-ended read and double-ended write are interesting as they can be optimised for read and write separately with minimal area overhead compared to a standard 6T cell.

Still, the traditional 6T cell must not necessarily be phased out yet. An implementation with SPHVT transistors is still stable enough in $90nm$, even when all transistors have equal width and length ($W = 200nm$, $L = 80nm$). A quantitative exploration of the impact of the cell bias voltages on cell stability and write-ability indicates that the available margins increase significantly when different bias voltages are applied for read and write operations. With such bias schemes, the 6T cell operation is assisted by the periphery. Such assistance allows the use of small 6T cells with rather poor intrinsic specifications and it enables the continued usage of traditional 6T cells in technology nodes in which traditional periphery cannot provide yielding minimal size memory cells.

Chapter 3 covers all topics that are related to the **local memory matrix**. A representative local matrix is sketched, and different implementations for word

line drivers are discussed. Special attention is paid to practical bias schemes that improve the read stability and write-ability of the 6T SRAM cell. A review of bias schemes that have been discussed in the literature indicates that these techniques are becoming mainstream as technology moves beyond $65nm$.

Different implementations of the local bit line periphery are presented. A good bit line hierarchy significantly reduces the impact of the cell current I_{read} and the cell's bit line capacitance $C_{BL,cell}$ on the access time and energy consumption of the memory.

An in-depth treatment of dynamic read stability is provided. A review of the recent literature on this topic shows that this technique, which we first introduced in [Cos06], receives a lot of attention now. The traditional metric for cell stability during the read operation, SNM_{read}, is a pessimistic estimate: cells with a negative SNM_{read} can still be read without suffering from a read upset. Examples indicate that the difference between the failure rate that is predicted by SNM_{read} and the real failures is very large when short buffered local bit lines are used. A novel metric to quantify a cell's dynamic read stability, SNM_{tran}, is introduced in this text. It expresses dynamic read stability as a value which can be directly compared with SNM_{read} and SNM_{hold}. Several design options can be explored to improve the cell's dynamic read stability: the number of cells per bit line, the word line rise time and the bit line precharge voltage.

Chapter 4 addresses another critical building block of low-power SRAM memories: the global read **sense amplifiers**. The important concept of sense amplifier offset is discussed and an overview of "noise" sources that negatively affect the sense amplifier operation is provided: transistor mismatch, temperature, bias voltages, aging and thermal noise. The ecosystem in which the SA operates is described and the important specifications for SAs are explained: the required differential input signal ΔV_{min}, the energy consumption per operation E_{SA} and the SA delay.

A practical simulation-based analysis method for strobed voltage mode SAs is outlined. This analysis includes transistor mismatch and thermal noise. This analysis is applied to several implementations of strobed voltage-mode SAs. A simple equation to estimate the noise performance of strobed SAs with regenerative feedback is validated against simulation results. The drain-input latch-type sense amplifier is clearly the best sense amplifier in the comparison. Only when noise performance or input current is a dominant problem, the pulsed current source SA which we re-introduced is the best alternative. In the example 90nm technology, the input transistors of a drain-input latch-type SA must be $1080nm$ wide (9x $120nm$) to be able to sense a $100mV$ complementary input signal. With correct sizing, SA offset is dominated by the input transistors. This SA has a dynamic energy consumption of $8.3fJ$ at $1V$. Figure 4.13 shows that voltage scaling down to $450mV$ does not introduce problems when SPLVT transistors are used. At $450mV$, the dynamic energy consumption is reduced from $8.3fJ$ to $1.44fJ$.

Chapter 5 discusses **sense amplifier calibration**, a family of techniques that

overcome the traditional limitations on SA performance. These techniques can reduce the required differential input signal ΔV_{min} with a factor of 5 and more for a fixed SA energy consumption. Three calibration techniques are discussed: SA redundancy, SA tuning and asymmetric SA redundancy. SA redundancy was introduced first in [Ver08b] to reduce the SA failure rate for a fixed SA area. We introduced SA tuning in [Cos08], asymmetric SA redundancy was not disclosed previously. Both SA tuning and asymmetric SA redundancy are significantly more effective at reducing E_{SA} and ΔV_{min} than SA redundancy.

Chapter 6 discusses the **transfer of the data bits** between the memory cell and the memory output interface during read and from the memory input interface to the memory cell during write. An overview of bit line structures and signalling types to implement these transfers is provided. For typical implementations of the read operation, both the energy and delay of the data transfer are proportional to ΔV_{min} of the global read sense amplifiers. The bit line write energy in the traditional scheme is 5 to 20 times larger than the bit line read energy. Several techniques to reduce the write energy are discussed. The most interesting approach is to add local low-swing write receivers that amplify the small-swing signal from the global bit lines to a full-swing signal on the local bit lines. Such an approach requires subdivided bit lines.

Chapter 7 outlines the **global control circuits**, which consist of the circuits for **timing** control and of the **decoder**. During a memory access, several timing events occur, such as the activation of the word line and the activation of the global bit line precharge. To ensure correct operation, these events must satisfy several delay constraints. This chapter describes the typical timing events and the associated delay constraints. Many different approaches can be used to ensure the correct order of events, so several choices must be made. First, timed delays can be generated by local timing circuits, or they can be generated by global timing circuits, in which case they still need to be transmitted to the correct local block. Second, more or fewer external clock events can be used as inputs for the timing control circuits. Third, three methods can be used to ensure that the correct operation order between events is respected: the simulator-based approach, the replica-based approach and the calibration-based approach.

The decoder must activate the correct block and word. Different organisations for the decoder are compared and the implementation of the decoder gates is illustrated.

A large part of the energy for the decoder and timing is associated with the transmission of signals across the memory matrix. Reduced-swing transmission is an interesting yet challenging option.

Chapter 8 presents the two **prototype memories** that have been designed, fabricated and measured. They demonstrate the feasibility of the novel techniques as outlined in section 1.4 (except asymmetric SA redundancy).

The second prototype supports different operation mode. In the $1.2V$ mode, the

maximum frequency is $850MHz$, $\frac{E}{WL} = 263\frac{fJ}{bit}$ and $\frac{E}{WL\sqrt{size}} = 725\frac{aJ}{bit^{1.5}}$. In the $0.6V$ mode,the maximum frequency is $240MHz$, $\frac{E}{WL} = 84\frac{fJ}{bit}$ and $\frac{E}{WL\sqrt{size}} = 233\frac{aJ}{bit^{1.5}}$. Selective voltage scaling and calibration-based timing enable an efficient run-time trade-off between energy and delay.

A comparison of the performance of our prototypes with that of other state-of-the-art designs is provided in section 8.4. Our second prototype clearly outperforms other state-of-the-art memories when a low energy consumption per operation is required at access speeds between $100MHz$ and $850MHz$.

9.2 Conclusions

Portable electronic devices have become a key part of everyday life. Just think about portable media players, gaming consoles and smartphones. Consumers expect ever more performance and features from these devices, while the battery capacity hardly improves. A low energy consumption per function is hence one of the most fundamental requirements in their design. These devices typically operate with a clock frequency between $100MHz$ and $1GHz$.

In most applications, memories consume a significant part of the total energy budget. After optimizations have been applied, most applications have strong data locality. This increases the relative importance of the small memories that are closest to the processor to such an extent that they often dominate the energy consumption. Hence, energy efficient implementations for these relatively small memories are a key requirement to enable further extensions of the capabilities of these devices.

To enable exciting new applications, a further radical reduction of energy consumption per operation is needed. Three tracks can contribute to this energy reduction: improvements in process technology, in system-level design and in circuit-level design.

For some applications, further miniaturisation can account for a part of this reduction, but only if new circuit and system solutions are devised to cope with the increasing variability and leakage currents.

At system-level, co-optimisation of the algorithms and the hardware system can often result in a huge reduction in the energy consumption. The maximal benefits from this approach are obtained when application-domain specific memories are employed. These memories can make the optimal trade-off between the specifications (e.g. energy, delay, area, yield and special features) for the application domain.

This work explores the third track. It improves the circuit-level design of relatively small SRAM memories in deep sub-micron technologies. The main design target is a low energy per operation at clock frequencies between $100MHz$ and $1GHz$. Deep sub-micron technologies introduce several new design challenges: large uncorrelated intra-die variations, leakage currents and wire dominance. In this work, these new issues are treated as first-order design considerations.

To increase the design freedom, the area consumption is considered a lesser design constraint. In many applications, trading some area for a large reduction in energy consumption or a significantly improved behaviour in deeply scaled technologies is acceptable for the targeted small memories.

Several new circuit techniques to reduce the energy per operation and to reduce the impact of leakage and transistor variations have been discussed in this text.

Short buffered bit lines with low-swing extended GBL reduce the dynamic energy consumption and improve the speed of the data transfer from memory cell to memory IO. This bit line structure reduces the impact of the cell read current on the total memory delay. This eases the use of high-threshold memory cells to reduce leakage and it improves the resilience against intra-die variations.

This bit line structure also allows the use of **dynamic read stability** as a method to prevent destruction of cell data during read, which increases the available cell margins. The available cell margins can also be increased significantly by applying different cell bias voltages during read and write operations. A lower cell supply voltage during write for example improves the write-ability. Such bias schemes allow the use of small 6T cells with rather poor intrinsic specifications and it enables the continued usage of traditional 6T cells in technology nodes in which traditional periphery cannot provide yielding minimal size memory cells.

Improved sense amplifiers further reduce the dynamic energy consumption and improve the speed of the data transfer from memory cell to memory IO. The energy versus offset performance of normal sense amplifiers is fundamentally limited by Pelgrom's law. This can be circumvented by calibration techniques, such as sense amplifier redundancy and **sense amplifier tuning**.

These techniques have been used in two prototype memories. The second prototype provides several operation modes. **Selective voltage scaling** and **calibration-based timing** enable an efficient run-time trade-off between energy and delay, from $8.4pJ$ per access at $850MHz$ to $2.7pJ$ per access at $240MHz$. Such **run-time trade-offs** provide flexibility to system designers. The prototypes outperform other state-of-the-art memories when a low energy per access must be combined with an access frequency above $100MHz$.

Proper circuit design can reduce the energy consumption of SRAM memories and can reduce the impact of new challenges such as transistor variations, leakage and wire dominance. Several techniques have been validated in this work, with encouraging results. Large further improvements are certainly still possible. Some interesting research directions are outlined in the next section.

9.3 Future work

Power supply management

In our prototypes, several external supply voltages are used. Some of these can easily be avoided, yet others are fundamental to obtain reliable, fast, low power memories in DDSM technologies. Multiple supply voltages can however be detrimental for the cost and complexity of integration in the total system. The cost of power supply generation should hence be taken into account during the memory design. This cost depends on which supplies are already needed in the system for other components. For supply voltages which are not yet available in the system, integrating a corresponding supply voltage generator together with the memory might be the most cost-effective solution. It certainly enables a reasonable comparison of the effective cost of different solutions. This is for example important when comparing two options to improve cell write-ability: making Vdd_{cell} float, or connect it to (an externally provided) lower supply voltage.

Some crucial supply voltages are listed below:

- The supply voltage for the **decoder**. This voltage should be reduced when delay slack is available to reduce the dynamic energy consumption. When low threshold transistors are used in these circuits, a supply voltage of $500mV$ is not unreasonable. The energy consumption from this supply is rather high. With static decoders, the consumed charge per cycle depends on the access pattern.

- The supply voltage for the **sense amplifiers** should behave similar to the decoder supply voltage. However, the trade-off between energy and delay is different, hence a different supply might be desirable.

- The cell supply voltage for **inactive cells** should be low to reduce the static power consumption. Currents from this supply are small in typical memories.

- The cell supply voltage for **active cells** cannot be reduced as much as that for the decoder, as the cell transistors have a higher threshold and are smaller. Different voltages for read and write operations might be considered to improve the cell margins. Energy consumption from this supply is small in typical memory organisations.

- The **word line voltage** should not be reduced much either. Energy consumption from this supply is small in typical memory organisations. Similar overdrive voltages are needed to control other pass transistors.

- The **global bit line precharge voltage** V_{prech} is a crucial supply voltage for the data transfer scheme with short buffered bit lines with low-swing extended GBL. Typical precharge voltages are between $50mV$ and

$300mV$. In this work, an ideal series regulator is assumed as generator for this supply. The corresponding energy consumption is $E_0 = C_{switched} \cdot V_{swing} \cdot Vdd$. With an ideal DC-DC convertor as voltage generator, the energy consumption would be $E_{DC-DC} = C_{switched} \cdot V_{swing} \cdot V_{prech}$ An ideal voltage generator can hence reduce the energy consumption of the data transfer with a factor of 2 ($Vdd = 600mV$, $V_{prech} = 300mV$) to 20 ($Vdd = 1V$, $V_{prech} = 50mV$). Whether the use of a DC-DC convertor with reasonable efficiency is feasible or not depends on the required accuracy of V_{prech}. Complementary signalling schemes are more robust to variations in this voltage than single-ended signalling schemes.

- In some charge-recycling schemes, a $Vdd/2$ supply voltage is used. The exact voltage level is often not very crucial (e.g. a range of $\pm 50mV$ is acceptable), and the current consumption is typically very small thanks to the charge recycling.

In future design explorations, selective voltage scaling should receive even more attention. For actual implementations, the cost and opportunity of adding additional supply voltages should be carefully weighted. Where appropriate, voltage convertors with dedicated specifications should be integrated alongside (or even inside) the memory.

Improved timing and decoder

The global control circuits consume a large fraction of the total available access time. When the word length is relatively small, the energy consumption of these circuits is very relevant as well. In larger memory matrices, a better reduced-swing transmission technique for activation signals and for decoder signals would significantly reduce their energy consumption.

Calibration techniques

This work employs sense amplifier calibration and timing calibration to improve the energy-delay-yield performance trade-off. Other calibration techniques should be included as well in future design explorations.

A well-known other calibration method is cell redundancy. Rather straightforward implementations can relax the cell requirements from about 6σ ($fr = \frac{1}{10^9}$) to about 4σ ($fr = \frac{1}{30000}$) with little energy, delay and area overhead.

To further relax the cell requirements with traditional redundancy schemes to e.g. 3σ is significantly more expensive in terms of energy, delay and area overhead . Calibrated cell bias voltages and other operation knobs can be significantly more cost-effective to implement and can equally well correct parametric failures.

As calibration techniques become more beneficial in advanced DSM designs, a standard framework for calibration control becomes a necessity to lower the implementation costs and risks of these calibration techniques and hence to ease the adoption by industry. IMEC's "standardized fine-grained knobs and

monitors (SKM)" is an example of such a framework. An example application is discussed in [Pap05].

Further scaling

An important question is whether and how SRAMs scale to technology nodes with smaller feature sizes, and which design challenges have to be addressed for such further scaling. Appendix D provides a preliminary assessment regarding this topic.

Performance estimates

This text discusses many alternative implementations for the different memory functions, which is very useful in its own right for circuit-level memory designers. However, it still requires a significant amount of work to obtain an estimate of the energy and delay trade-off that can be obtained for a memory of a given size and word length in a certain technology. This information is of crucial importance for a system designer though. When this information is compiled, an overview of the contribution of the different memory components to energy and delay is easily obtained as well. This is crucial information for circuit designers, as it shows them which components should be further improved.

A tool that provides reasonable estimates of the pareto-optimal energy and delay performance of a memory with a fixed topology should not be too hard to implement. Even if it is not a foolproof push-the-button solution, the results can be very informative.

Sleep and power down modes

Future memory designs should provide energy-efficient sleep and power down modes. The optimal implementation will depend to some extent on the properties of the application. A good combination of hardware and software control of the sleep states is probably required to obtain a solution that is sufficiently reusable yet efficient enough.

Appendix A

Practical methods to obtain SA noise performance

As discussed in chapter 4, the noise performance of a sense amplifier is expressed by $\sigma_{v_{noise}}$. This appendix gives an overview of how this parameter can be obtained by performing measurements, numerical simulations and analytical calculations.

Obtaining $\sigma_{v_{noise}}$ with measurements

Two ways to measure $\sigma_{v_{noise}}$ have been presented. In [Opr97], a closed loop noise measurement technique based on the output decision probability variation for inputs close to the transition point is used. $\sigma_{v_{noise}}$ is then obtained from the resulting duty cycle as function of the input voltage. A simpler method which is better suited to our needs is used in e.g. [Sch07, Nuz08]. This measurement approach will now be discussed in some detail.

A sweep of the input voltage is performed. For each input voltage V_{in}, N sensing operations are performed. Due to thermal noise, this is a probabilistic experiment. The number of sensing operations out of this N operations that results in a high output is called the success rate $N_1(V_{in})$.

Figure A.1 shows some important values used in the explanation. This graph was created with a matlab simulation. An underlying equivalent input referred Gaussian noise source with $\sigma_{v_{noise}} = 2mV$ was chosen. The SA offset voltage was set to $V_{offset} = 10mV$. For each step of the input voltage V_{in}, the results of $N = 100$ sensing operations were emulated.

If the probability of a metastable output is negligible, each individual sensing event is a Bernouilli experiment, with a probability p to obtain a high output and a probability $(1 - p)$ to obtain a low output. If $\sigma_{v_{noise}}$ is known, p can be calculated for each effective input voltage $V_{in,effective} = V_{in} - V_{offset}$ using the CDF of the Gaussian distribution, $p = normcdf(V_{in,effective}, \mu = 0, \sigma = \sigma_{v_{noise}})$.

Each $N_1(V_{in})$ is the number of successes in a sequence of N independent Bernouilli experiments, each with a success probability p, so $N_1(V_{in})$ follows a binomial distribution with mean $\mu_{N_1} = N \cdot p$ and standard deviation $\sigma_{N_1} = \sqrt{N \cdot p \cdot (1 - p)}$. The mean value of $N_1(V_{in})$ is the population average of $N_1(V_{in})$. This is the solid line in figure A.1.

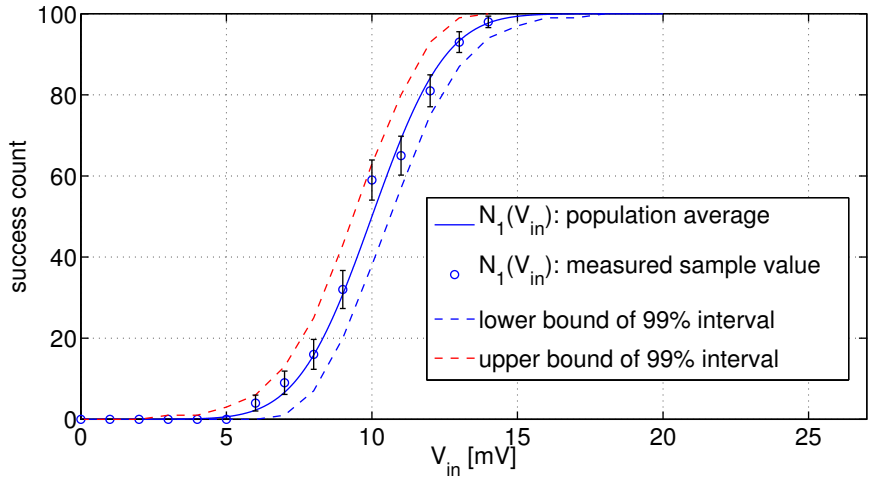


Figure A.1: Concepts used in the measurement of the input referred noise. [$N = 100$, $\sigma_{v_{noise}} = 2mV$, $V_{offset} = 10mV$]

When N sensing operations are performed for an input value V_{in}, N random samples are taken from v_{source}. This results in a single measured sample value $N_1(V_{in})$. The result of such a measurement is displayed by the small circles in figure A.1.

In the assumption that the strobed SA can be modelled as in figure 4.2, these measurement results allow to obtain an estimate of the parameters $\sigma_{v_{source}}$ and V_{offset}, for example by performing a (non-linear) least squares fit.

If the combination of N and the number of values of V_{in} that are measured are low, different measurement runs will result in significantly different estimates of the parameters. Figure A.2 shows the experimental CDF when performing multiple attempts to estimate $\sigma_{v_{source}}$ with a non-linear least squares fit. This provides a feel for the consistency of the estimation. The system under test is the example from figure A.1. At each of the indicated values of V_{in}, N measurements are performed. This exercise was repeated with different values of N. The resulting CDFs can be used to make an informed choice for the value of N to use in an actual measurement setup. A value of around 1000 is probably an acceptable choice in most situations.

When a model is obtained, it should be validated. A visual inspection is certainly a good starting point. However, to perform this check in an acceptable way, the amount of uncertainty of each measured value should be visualised as well. Two possible approaches for the estimation of this uncertainty will be discussed shortly.

The first approach does not use the fitted model. It uses only one measurement result (N sensing operations) for each input voltage V_{in} and the assumption of independent Bernouilli experiments to estimate the standard deviation of

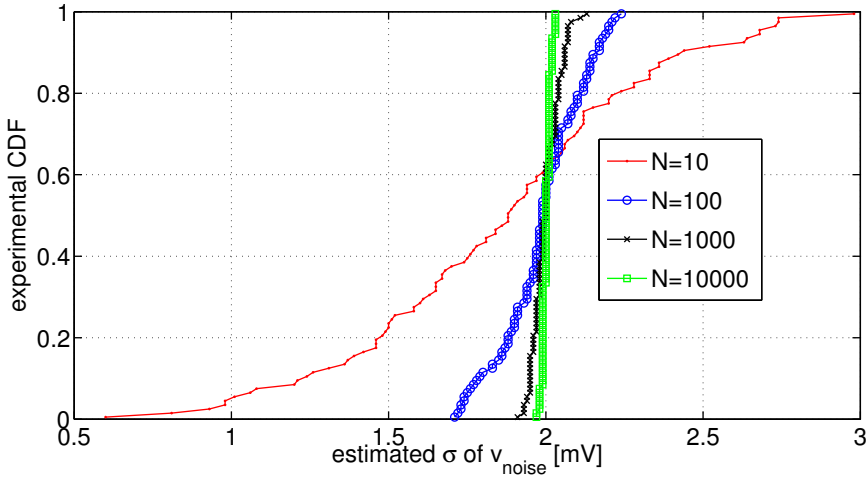


Figure A.2: CDF of estimated $\sigma_{v_{source}}$. [$\sigma_{v_{noise}} = 2mV$, measurements at V_{in} as shown as in figure A.1]

$N_1(V_{in})$ by replacing p in $\sqrt{N \cdot p \cdot (1-p)}$ with its estimate from measurement $\frac{N_1(V_{in})}{N}$. The vertical bars in figure A.1 indicate the estimated $[\pm 1 \cdot \sigma]$ intervals. This approach seems to be used in [Sch07].

The second approach starts from the fitted model. The model predicts the probability for each measurement outcome at each input voltage. In figure A.1, the smallest vertical interval that covers 99% or more of the measurement results at the given V_{in} is indicated by its upper and lower bound. If some of the measurement results are very unlikely according to the model, the model is most likely invalid. The reason for this deviation from the expected behaviour must then be discovered and addressed adequately.

Obtaining $\sigma_{v_{noise}}$ by numerical simulation

The same technique can also be used to estimate V_{offset} and $\sigma_{v_{noise}}$ based on numerical transient noise simulations. Transient noise simulations are now available in most major commercial simulators. For example, Eldo provides the command ".NOISETRAN". The important parameters are $FMIN$ and $FMAX$. When $FMIN = 0$, an algorithm with continuous noise sources is used[eld07]. With settings $FMIN = 0$ and $FMAX = 5e12$, the results seem to be rather self-consistent. Still, some verification would be desirable.

The transient noise simulation can be used to simulate N sensing events for each input voltage V_{in} in the desired sweep. This results in a simulated sample value N_1 for each input voltage. The model parameters V_{offset} and $\sigma_{v_{noise}}$ are then extracted in a post processing, as discussed above. A simulation for a typical SA using 20 steps of V_{in} and $N = 1000$ can be performed overnight on a recent

desktop computer. As with many other numerical simulations, blind faith in these simulations is probably not advisable.

Obtaining $\sigma_{v_{noise}}$ by analytical calculation

The model parameters can also be obtained (approximately) with analytical calculation. A very thorough and interesting elaboration can be found in [Nuz08]. The oversimplified outcome of their analysis is rather intuitive: $\sigma_{v_{noise}}$ is mainly influenced by the capacitances on the internal nodes ($\sim \sqrt{\frac{1}{C}}$) and by the gain from the input voltage to the voltage on the internal nodes at the onset of regenerative sensing. A much simpler estimate was used in [Opr97], but this still yielded a good accuracy for the SA under study.

Appendix B

Analytical analysis of strobed voltage-mode SAs

This appendix first provides an overview of the literature on the analytical analysis of strobed voltage-mode SAs. Then, first-order symbolic solutions are derived for a number of interesting sense amplifiers. Expressions will be derived for the (initial) nominal response and for the sensitivity of the offset voltage to mismatch in the threshold voltages of transistors. Especially the difference between a linear amplification phase and a regenerative amplification phase deserves some attention. Even though the resulting equations are not accurate enough to use in real designs, this symbolic analysis provides a significant amount of insight.

This appendix also provides details about the pre-amplification factor A_{pre}.

Overview of the literature

Both the nominal behaviour and the mismatch behaviour of strobed voltage-mode SAs can be approximated with analytical derivations.

Different approaches can be taken. Some authors use full-signal analysis, where they typically resort to rather simple device equations. Such approaches have been used to calculate optimal latching waveforms for strobed SAs and to calculate first-order approximations of offset voltage as function of transistor mismatch. In [Sar91] and [Wan82], a very mathematical viewpoint is taken. In [Kra89] and [Sin04], a less generic yet easier to grasp approach is used to calculate the optimal linear ramp rate of the SA enable signal to minimise SA offset. These two papers highlight that for the offset calculation, it is not enough to look only at $t = 0+$. In [Rya08], offset is analysed for a gate-input stacked latch-type SA that operates in the sub-threshold region.

Often, small-signal analysis is used to obtain formulas of interpretable complexity. In [Haj98], the analysis of the nominal response of the a drain-input latch-type SA is performed with small-signal parameters only. To account for transient changes in operating point, the small-signal parameters are made time-dependent. In [Wic04], a mix between small-signal analysis and very simple large-signal approximations is used to calculate the nominal delay of a gate-input stacked latch-type SA.

In [Nuz08], the noise performance of the gate-input stacked latch-type SA at the nominal supply voltage is analysed. Small-signal analysis is used. To account for

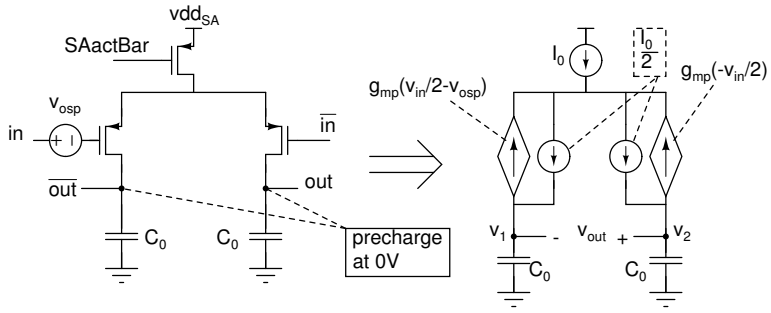


Figure B.1: The schematic of a differential input pair with precharged output and the corresponding first-order simplification.

the time-varying nature of the circuit, three operation phases were considered. For each phase, time averaged small-signal parameters were used.

First-order symbolic analysis

In this section, the (initial) nominal response and the sensitivity of the offset voltage to transistor threshold mismatch will be analysed with a simplistic first-order analysis for the most important SA input stages.

Differential input pair

Figure B.1 shows the schematic of a differential input pair with only a capacitive load. The output nodes are precharged to ground. In this simplified analysis, the schematic on the right will be used. The SA is enabled by lowering $SAactBar$. The top transistor acts as a current source and the two input transistors form a basic differential stage. Thanks to the precharge at 0V, they are indeed in saturation at the start of the operation. If there would not be any differential input voltage v_{in} and no transistor threshold mismatch v_{osp}, v_1 and v_2 would increase at the same rate, $\frac{dv_1}{dt} = \frac{dv_2}{dt} = \frac{I_0}{2 \cdot C_0}$. When v_{in} and v_{osp} are taken into account, we can apply kirchhoff's current law on nodes out and $\overline{out}$.

$$\frac{dv_1}{dt} C_0 = \frac{I_0}{2} - g_{mp} \left(\frac{v_{in}}{2} - v_{osp} \right)$$
$$\frac{dv_2}{dt} C_0 = \frac{I_0}{2} - g_{mp} \left(\frac{-v_{in}}{2} \right)$$

This pair of equations can be converted to two simpler equations: one equation describing the common mode response and one equation describing the differential mode equation by adding and subtracting the two equations. With $v_{out} = v_2 - v_1$, the equation for the differential mode is:

$$\frac{dv_{out}}{dt} C_0 = g_{mp} v_{in} - g_{mp} v_{osp}$$

With initial condition $v_{out}(0) = 0$, the differential output voltage is:

$$v_{out} = \frac{g_{mp}(v_{in} - v_{osp})}{C_0} \cdot t$$

If a certain gain $A = \frac{v_{out}}{v_{in}}$ is needed, and with $v_{osp} = 0$, the delay until this gain is reached is $t = delay = \frac{v_{out}}{v_{in}} \frac{C_0}{g_{mp}} = A\frac{C_0}{g_{mp}}$. The delay is linear with the required amplification.

Obviously, this trivial equation is only valid as long as the input transistors operate in the saturation region. When v_{in} and v_{osp} are perfectly balanced, $v_1 = v_2$ and

$$V_{sd} \leq V_{sg} - V_T$$

$$V_{top} - v_1 \leq V_{top} - V_{in,CM} - V_T$$

$$v_1 \geq V_{in,CM} + V_T$$

So when the common mode output reaches $V_{max} = V_{in,CM} + V_T$, the input transistors enter the linear region. After this point is reached, the differential signal will reduce, as will be explained further on. The question of interest is: how much gain can be obtained from a small differential input voltage v_{in} to the differential output voltage v_{out} before V_{max} is reached? The linear region is reached at time $t_{lin} = \frac{Q}{I} = \frac{2C_0 \cdot V_{max}}{I_0}$. At this time, the differential signal is $v_{out} = \frac{g_{mp}v_{in}}{C_0} \cdot t_{lin} = \frac{2g_{mp}V_{max}}{I_0}v_{in}$. In strong inversion, $g_{mp} = \frac{2I_{sd}}{V_{sg}-V_T} = \frac{I_0}{V_{sg}-VT}$, and we obtain $v_{out} = \frac{2(V_{max})}{V_{top}-V_{in,CM}-V_T}v_{in}$.

Consider for example the global read SAs in our 90nm design [Cos08]. The global bit lines are precharged low, so lets assume $V_{in,CM} = 0$. The available threshold voltages are between $0.25V$ and $0.4V$ and V_{top} can be anywhere between $0V$ and $1V$. However, when V_{top} is below about $V_T + 70mV$, the input transistors do not operate in strong inversion and the derivation is not valid. With $V_{top} = 0.8V$, $V_{in,CM} = 0V$ and $V_T = 0.4V$, the maximum gain is only $A_{max} = \frac{v_{out}}{v_{in}} = 2$. When V_{top} is reduced to $0.6V$, the maximum gain is 4.

The finite output impedance of the input transistors also puts a limit on the gain that can be obtained. Figure B.2 shows the small-signal schematic. The transistor output impedance used to be $r_0 = \frac{V_E L}{I_{ds}}$, but do not rely on this for the 90nm devices.

The differential equation becomes

$$\frac{dv_{out}(t)}{dt}C_0 = g_{mp}(v_{in} - v_{osp}) - \frac{v_{out}(t)}{r_0}$$

With initial condition $v_{out}(0) = 0$, the differential output voltage is:

$$v_{out}(t) = g_{mp}r_0(v_{in} - v_{osp})\left(1 - e^{-\frac{t}{r_0 C_0}}\right)$$

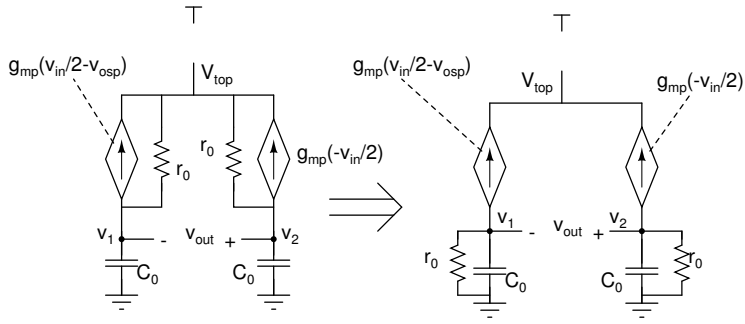


Figure B.2: The small-signal schematic of a differential input pair with precharged output, taking the finite transistor output impedance into account. The simplification to ground can be used for the analysis of the differential output voltage.

For small time values, this response is similar to the one without r_0. For large time values, it evolves towards the traditional gain $g_{mp}r_0$. For $V_{ds} = 1V$ and $V_{gs} = 1V$, minimal length SPLVT transistors in 90nm technology have a gain of 5, which is probably enough for a linear pre-amplification. However, in the linear region, this product becomes much smaller than one, effectively attenuating the input signal. The linear operation region must be avoided at all cost until the sensitive part of the sensing operation is completed.

The gain could be increased by increasing L or by decreasing $V_{gs}-V_T$. How does the choice of V_{gs} and L affect the SA delay? In many cases, the capacitance C_0 is dominated by the input transistor, and $C_0 \approx W \cdot L \cdot C_{ox}$. In saturation $g_{mp} = \frac{2I_{ds}}{V_{gs}-V_T}$, and $I_{ds} = K\frac{W}{L}(V_{gs}-V_T)^2$, so:

$$delay = A\frac{C_0}{g_{mp}} \approx A\frac{C_{ox}WL}{\frac{2K\frac{W}{L}(V_{gs}-V_T)^2}{V_{gs}-V_T}} = A\frac{C_{ox}L^2}{K(V_{gs}-V_T)}$$

Differential input pair with cross-coupled load

In figure B.3, the differential input pair is loaded with cross-coupled NMOS latches. Assuming constant small signal parameters, the differential output voltage is described by equation

$$\frac{dv_{out}(t)}{dt}C_0 = g_{mp}(v_{in}-v_{osp}) + g_{mn}(v_{out}(t)-v_{osn}) - \frac{v_{out}(t)}{R}$$

$$v_{out}(t) = Rg_{mp}(v_{in}-v_{osp}-\frac{g_{mn}}{g_{mp}}v_{osn})\left(e^{\frac{g_{mn}R-1}{RC_0}t}-1\right)\frac{1}{g_{mn}R-1}$$

Notice that this is a positive exponential due to the positive feedback. The delay to obtain a certain amplification $A = \frac{v_{out}}{v_{in}}$ depends logarithmically on A.

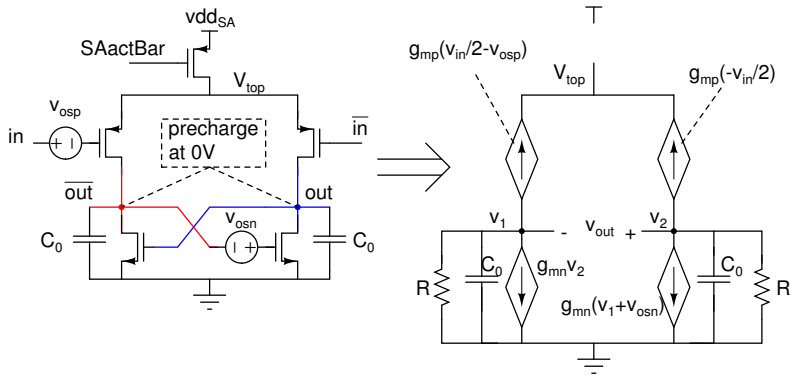


Figure B.3: The small-signal schematic of a differential input pair with precharged output and cross-coupled NMOS load.

For small values of x, $exp(x) \approx 1 + x$, and this equation can be simplified to

$$v_{out}(t) = \frac{g_{mp}}{C_0}(v_{in} - v_{osp} - \frac{g_{mn}}{g_{mp}}v_{osn})t$$

Another interesting simplification is obtained for $R = \infty$.

$$v_{out}(t) = \left(\frac{g_{mp}}{g_{mn}}(v_{in} - v_{osp}) - v_{osn}\right)\left(e^{\frac{g_{mn}}{C_0}t} - 1\right)$$

The minimal required differential input voltage v_{in} that is needed to balance out the transistor mismatch v_{osn} and v_{osp} is the offset voltage of the SA. In this case, it is $V_{offset} = v_{osp} + \frac{g_{mn}}{g_{mp}}v_{osn}$. The addition of the NMOS transistors has increased the SA offset. However, the impact can be minimal if g_{mp} is significantly larger than g_{mn}. If this would be a static situation, making g_{mn} small would increase the delay. In practice, the output nodes are precharged to ground, so at the beginning of the sensing operation, $g_{mn} \approx 0$. Only when the common mode voltage of the output reaches the threshold voltage V_{Tn} of the NMOS transistors, g_{mn} becomes significant. By this time, the PMOS input pair has already amplified the input signal with a factor $A_{pre} = \frac{2(V_{max})}{V_{top} - V_{in,CM} - V_{Tp}}$, with $V_{max} \approx V_{Tn}$. This basically reduces the impact of v_{osn} on V_{offset} with a factor A_{pre}.

Drain-input latch-type SA

Figure B.4 shows the simplified small-signal schematic of the drain-input latch-type SA. Before activation, the input signal is transferred to the internal nodes. This implementation can only be used when the common input signal $V_{in,cm}$ is close to ground. Otherwise, a switched current source must be added at the bottom as well. In this section, $V_{in,cm}$ is assumed to be low, and the schematic from figure B.4 is used.

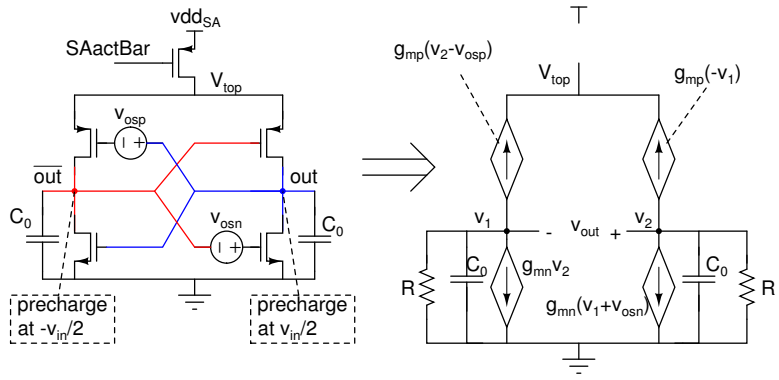


Figure B.4: The small-signal schematic for a drain-input latch-type SA.

After activation, the differential voltages are described by

$$\frac{dv_{out}(t)}{dt}C_0 = g_{mp}(v_{out}(t) - v_{osp}) + g_{mn}(v_{out}(t) - v_{osn})$$

The solution to this equation with the initial condition $v_{out}(0) = v_{in}$ is

$$v_{out}(t) = e^{\frac{g_{mp}+g_{mn}}{C_0}t} \cdot \left(v_{in} - \frac{g_{mp}v_{osp} + g_{mn}v_{osn}}{g_{mp} + g_{mn}}\right) + \frac{g_{mp}v_{osp} + g_{mn}v_{osn}}{g_{mp} + g_{mn}}$$

The positive feedback again leads to a positive exponential. After sufficient time, the last term can be neglected. The offset voltage for this SA is

$$V_{offset} = \frac{g_{mp}v_{osp} + g_{mn}v_{osn}}{g_{mp} + g_{mn}} \tag{B.1}$$

Again, the impact of v_{osn} on V_{offset} can be reduced by using the PMOS transistors as a pre-amplifier. The positive feedback of the PMOS transistors makes it much easier to obtain a larger A_{pre}.

Appendix C

Important technology parameters

This appendix provides a brief overview of the $90nm$ technology that has been used for the examples in this text.

Transistors

Currents and threshold voltage

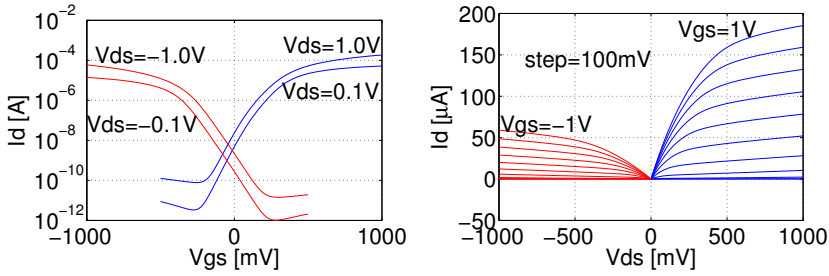
Table C.1 provides an overview of the transistors that are available in the example technology. The on current of the NMOS transistors is about 2.7 times larger than that of the PMOS transistors. At a supply voltage of $1V$, the leakage of HVT transistors is 100 times smaller than that of LVT transistors, while the on current is about 40% smaller. Figure C.1 shows the drain current of a SPLVT transistor.

Transistor variations

In this work, transistor variations are included in the simulation netlist as shown in figure C.2. Only variations in the threshold voltage V_T and in β are included. ΔV_T is added as an independent voltage source before the gate of the transistor. The variations in β are expressed as $\frac{\Delta\beta}{\beta_0}$ which can be implemented as a current controlled current source between drain and source. This implementation seems adequate for this $90nm$ technology. A possible problem of this implementation for future technologies is that the voltage source for ΔV_T alters the gate current, which is not a physical effect. Adding the voltage and current source to all

	LVT		SVT		HVT	
	NMOS	PMOS	NMOS	PMOS	NMOS	PMOS
$V_{T,lin}$ [mV]	270	-240	330	-280	460	-400
$V_{T,sat}$ [mV]	160	-140	240	-180	370	-300
I_{on} $[\frac{\mu A}{\mu m}]$	800	290	650	280	480	180
I_{off} $[\frac{nA}{\mu m}]$	60	40	5	10	0.3	0.4

Table C.1: Parameters for the transistors in the example technology at room temperature and at $1V$. $V_{T,lin}$ is defined at $Vds = 50mV$, $V_{T,sat}$ at $Vds = 1V$. All values are for transistors with minimal length.



(a) (b)

Figure C.1: Transistor drain current for SPLVT NMOS and PMOS transistors with $W = 200nm$ and $L = 80nm$. ($Vsb = 0$).

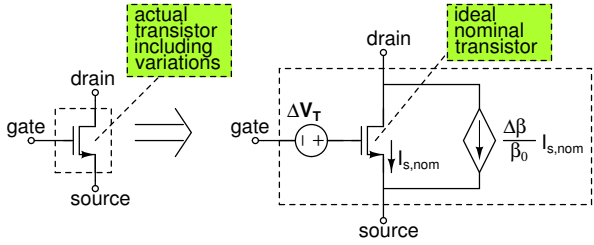


Figure C.2: Setup to include transistor variations in simulations.

transistors in the simulation of a memory slice with 10000 transistors increases the simulation time with 30%, which is rather acceptable.

The uncorrelated stochastic intra-die variations are modeled by selecting each ΔV_T from a Normal distribution with 0 mean and $\sigma_{V_T} = \frac{A_{\Delta V_T}}{\sqrt{2}\sqrt{W \cdot L}}$, and each $\frac{\Delta\beta}{\beta_0}$ from a Normal distribution with 0 mean and $\sigma = \frac{A_{\frac{\Delta\beta}{\beta}}}{\sqrt{2}\sqrt{W \cdot L}}$. All parameters are assumed to be entirely uncorrelated.

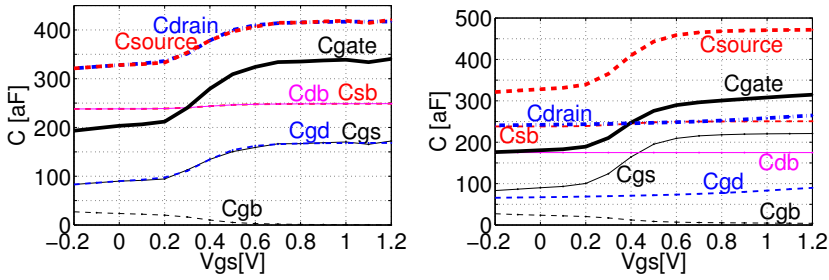
Transistor capacitances

- SPHVT NMOS: see figure C.3
- SPHVT PMOS: see figure C.4

Wire resistance and capacitance

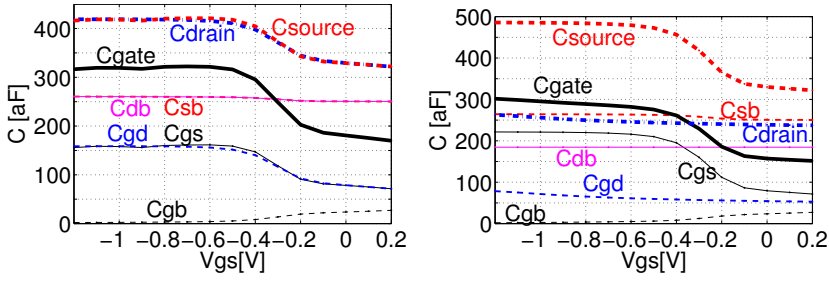
Wire parameters for $90nm$ technology (lowest metal layers):

- sheet resistance: $0.1\Omega/\square$
- capacitance: between $0.1\frac{fF}{\mu m}$ and $0.25\frac{fF}{\mu m}$



(a) $Vds = 0mV$ (b) $Vds = 1200mV$

Figure C.3: Transistor capacitances for a SPHVT NMOS transistor with $W = 200nm$ and $L = 80nm$. ($Vsb = 0$).



(a) $Vds = 0mV$ (b) $Vds = -1200mV$

Figure C.4: Transistor capacitances for a SPHVT PMOS transistor with $W = 200nm$ and $L = 80nm$. ($Vsb = 0$).

Appendix D

Consequences of further scaling

This appendix assesses the consequences of further technology scaling on SRAM design issues and SRAM performance. A first section outlines the technology assumptions that are used, The next section outlines the consequences of scaling under these assumptions. To avoid excessive complexity, only stochastic intra-die variations are considered. Die-to-die variations, systematic variations and the impact of aging and temperature are ignored.

Assumptions

Transistor scaling

The assessment is based on the technology requirements as projected by the 2007 edition of the ITRS roadmap. Table D.1 provides an overview of the expected general trends. Notice that three clearly distinct types of transistors are considered: transistors for high performance logic (V_T≈$100mV$), transistors for low standby power (V_T≈$500mV$) and transistors for low operating power (V_T≈$250mV$ and reduced Vdd). In the $90nm$ node, a single technology typically provides transistors with V_T values in a $200mV$ range. It is not clear whether this range will increase or not with scaling.

In the ITRS predictions, the gate leakage for a minimal-size transistor follows the same scaling trend as the drain/source leakage, $I_{leak} \sim \lambda$, and has a very similar magnitude.

According to the table, the dynamic energy consumption per minimal-sized transistor reduces as $C \cdot Vdd^2 \sim \lambda^{1.75}$, while the leakage power decreases as $I_{leak} \cdot Vdd \sim \lambda^{1.25}$. The balance between dynamic and leakage energy does not shift significantly if the operation frequency is kept constant. The nominal gate delay reduces as $delay \sim \lambda$.

The scaling trend for the transistor threshold mismatch is assumed to be $A_{\Delta V_T} \sim \sqrt{\lambda}$, which was considered a practical value for planar bulk transistors in [Ito09]. We select $A_{\Delta V_T} = 3.5mV \cdot \mu m$ for $90nm$ as base value. The impact of β variations is not considered. The corresponding mismatch values are shown in table D.2.

High-performance logic						
$\lambda\ [nm]$	68	45	32	28	25	trend
$Vdd\ [V]$	1.1	1.0	0.9	0.9	0.9	$\sim \sqrt[4]{\lambda}$
$V_{T,sat}\ [mV]$	134	100	100	100	100	constant
$I_{sd,leakage}\ [\frac{\mu A}{\mu m}]$	0.34	0.6	0.6	0.6	0.6	constant
$I_{sd,leakage}\ [\frac{nA}{2\cdot\lambda}]$	46	54	38	33	30	$\sim \lambda$
$I_{dsat}\ [\frac{\mu A}{\mu m}]$	1211	1807	2000	2000	2000	$\sim \frac{1}{\sqrt{\lambda}}$
$I_{dsat}\ [\frac{\mu A}{2\cdot\lambda}]$	165	163	128	112	100	$\sim \sqrt{\lambda}$
$C_{gate}\ [\frac{aF}{\mu m}]$	710	840	660	580	540	$\sim \sqrt[4]{\lambda}$
$C_{gate}\ [\frac{aF}{2\cdot\lambda}]$	97	76	42	32	27	$\sim \lambda^{1.25}$
$tau\ [ps]$	0.64	0.46	0.3	0.26	0.24	$\sim \lambda$

Low standby power						
$\lambda\ [nm]$	68	45	32	28	25	trend
$Vdd\ [V]$	1.1	1	0.9	0.9	0.85	$\sim \sqrt[4]{\lambda}$
$V_{T,sat}\ [mV]$	534	535	400	400	400	constant
$I_{sd,leakage}\ [\frac{pA}{\mu m}]$	30	30	30	30	30	constant
$I_{sd,leakage}\ [\frac{pA}{2\cdot\lambda}]$	4.1	2.7	1.9	1.7	1.5	$\sim \lambda$
$I_{dsat}\ [\frac{\mu A}{\mu m}]$	465	528	600	700	700	$\sim \frac{1}{\sqrt{\lambda}}$
$I_{dsat}\ [\frac{\mu A}{2\cdot\lambda}]$	63	48	38	39	35	$\sim \sqrt{\lambda}$
$C_{gate}\ [\frac{aF}{\mu m}]$	710	840	660	580	540	$\sim \sqrt[4]{\lambda}$
$C_{gate}\ [\frac{aF}{2\cdot\lambda}]$	97	76	42	32	27	$\sim \lambda^{1.25}$
$tau\ [ps]$	1.68	1.59	0.99	0.75	0.66	$\sim \lambda$

Low operating power						
$\lambda\ [nm]$	68	45	32	28	25	trend
$Vdd\ [V]$	0.8	0.7	0.7	0.6	0.6	$\sim \sqrt[4]{\lambda}$
$V_{T,sat}\ [mV]$	290	260	200	200	200	constant
$I_{sd,leakage}\ [\frac{nA}{\mu m}]$	10	10	10	10	10	constant
$I_{sd,leakage}\ [\frac{pA}{2\cdot\lambda}]$	1360	900	640	560	500	$\sim \lambda$
$I_{dsat}\ [\frac{\mu A}{\mu m}]$	563	682	750	810	850	$\sim \frac{1}{\sqrt{\lambda}}$
$I_{dsat}\ [\frac{\mu A}{2\cdot\lambda}]$	77	62	48	45	43	$\sim \sqrt{\lambda}$
$C_{gate}\ [\frac{aF}{\mu m}]$	710	840	660	580	540	$\sim \sqrt[4]{\lambda}$
$C_{gate}\ [\frac{aF}{2\cdot\lambda}]$	97	76	42	32	27	$\sim \lambda^{1.25}$
$tau\ [ps]$	1.19	0.86	0.62	0.43	0.38	$\sim \lambda$

Table D.1: Information compiled from the 2007 edition of the ITRS roadmap. The rows that are expressed per $2 \cdot \lambda$ provide approximate values for the smallest transistor without T-bone structure.

$\lambda\ [nm]$	90	65	45	32	22	16	10
$A_{\Delta V_T}\ [mV \cdot \mu m]$	3.5	3.0	2.5	2.1	1.7	1.5	1.2
$\sigma_{V_T,2\lambda}\ [mV]$	19	23	27	33	39	46	58

Table D.2: Guestimates for the threshold voltage mismatch.

Wire scaling

Simple idealised uniform scaling for all wire dimensions is assumed. The wire resistance per square scales as $R_\square \sim \frac{1}{\lambda}$, the wire capacitance per micron stays constant, while the wire capacitance per square for a wire of minimal width scales as $C_\square \sim \lambda$. Under this assumption, the delay caused by wire resistance is constant. Because the gate delay does reduce, wire delay becomes a significant issue for fast memory designs. If the memory speed is kept constant with scaling, e.g. by reducing the supply voltage, the wire resistance remains a secondary issue for memories with sizes around $128Kbit$.

For a fixed memory architecture, the energy consumption in wire capacitances scales as $C \cdot Vdd^2 \sim \lambda^{1.5}$. As long as the wire resistance does not come into the picture, the balance between transistor and wire energy changes only slightly.

Consequences of scaling

Logic at nominal supply voltage

The nominal performance of static CMOS logic circuits that operate at the nominal supply voltage scales as

- $E_{dynamic} \sim C \cdot Vdd^2 \sim \lambda^{1.75}$
- $P_{leakage} \sim I_{leakage} \cdot Vdd \sim \lambda^{1.25}$
- $Delay \sim \frac{C \cdot Vdd}{I_{dsat}} \sim \lambda$

However, with scaling, σ_{V_T} increases and the overdrive voltage $Vdd - V_T$ decreases, which degrades the actual performance as compared to the nominal performance.

A larger σ_{V_T} increases the average leakage power. Only the drain/source subthreshold leakage is considered here. If $I_{leakage}$ is given by

$$I_{leakage} = I_0 \cdot e^{\frac{-V_T}{nkT/q}}$$

then $I_{leakage}$ follows a log-normal distribution with

$$\begin{aligned} \mu &= \ln I_0 + \frac{-V_{T,0}}{nkT/q} \\ \sigma &= \frac{\sigma_{V_T}}{nkT/q} \end{aligned}$$

$\lambda\ [nm]$	90	65	45	32	22	16	10
$A_{\Delta V_T}\ [mV \cdot \mu m]$	3.5	3.0	2.5	2.1	1.7	1.5	1.2
$\sigma_{V_T,2\lambda}\ [mV]$	19	23	27	33	39	46	58
$\left(\frac{I_{mean}}{I_{median}}\right)_{2\lambda}$	1.27	1.33	1.40	1.51	1.63	1.78	2.06
High-performance logic							
$Vdd\ [V]$	1.2	1.1	1.0	0.9	0.9	0.8	0.7
$V_T\ [mV]$	130	100	100	100	100	100	100
$Vdd - V_T\ [mV]$	1070	1000	900	800	800	700	600
$Vdd - V_T - 6\sigma_{V_T}\ [mV]$	950	860	740	600	570	420	250
$\frac{D_{6\sigma}}{D_{nominal}};\ \alpha = 1$	1.12	1.16	1.22	1.33	1.41	1.65	2.38
$\frac{D_{6\sigma}}{D_{nominal}};\ \alpha = 2$	1.25	1.35	1.49	1.77	2.0	2.73	5.67
Low standby power							
$Vdd\ [V]$	1.2	1.1	1.0	0.9	0.8	0.75	0.7
$V_T\ [mV]$	500	500	400	400	400	400	400
$Vdd - V_T\ [mV]$	700	600	600	500	400	350	300
$Vdd - V_T - 6\sigma_{V_T}\ [mV]$	590	460	440	300	170	70	-50
$\frac{D_{6\sigma}}{D_{nominal}};\ \alpha = 1$	1.19	1.3	1.37	1.66	2.41	4.73	
$\frac{D_{6\sigma}}{D_{nominal}};\ \alpha = 2$	1.43	1.69	1.88	2.74	5.81	22.4	
Low operating power							
$Vdd\ [V]$	0.9	0.8	0.8	0.7	0.6	0.6	0.5
$V_T\ [mV]$	290	260	200	200	200	200	200
$Vdd - V_T\ [mV]$	610	540	600	500	400	400	300
$Vdd - V_T - 6\sigma_{V_T}\ [mV]$	500	400	440	300	170	120	-50
$\frac{D_{6\sigma}}{D_{nominal}};\ \alpha = 1$	1.23	1.34	1.37	1.66	2.41	3.23	
$\frac{D_{6\sigma}}{D_{nominal}};\ \alpha = 2$	1.51	1.8	1.88	2.74	5.81	10.4	

Table D.3: Impact of transistor variations on the performance of logic gates for devices with $L = \lambda$ and $W = 2\lambda$.

The mean and median leakage are given by

$$\begin{aligned} I_{median} &= e^{\mu} = I_0 \cdot e^{\frac{-V_{T,0}}{nkT/q}} \\ I_{mean} &= e^{\mu+\frac{\sigma^2}{2}} = I_{median} \cdot e^{\frac{1}{2}\left(\frac{\sigma_{V_T}}{nkT/q}\right)^2} \end{aligned}$$

With $nkT/q = 40mV$, the ratio between I_{mean} and I_{median} for transistors with $L = \lambda$ and $W = 2 \cdot \lambda$ increases from 1.3 in $90nm$ to 2.1 in $10nm$. The impact of this on the total chip leakage is probably not yet significant.

A first-order approximation of the gate delay is

$$D = \frac{C \cdot Vdd}{I_{on}}$$

If the transistor operates in the saturation region, $I_{on} \approx K \cdot (V_{gs} - V_T)^{\alpha}$. For this analysis, V_{gs} is assumed to be Vdd. α depends on the technology and the bias conditions, but is confined between 1 and 2. Variations in the on current due to variations in V_T result in delay variations. The ratio between the delay caused by a transistor with a threshold voltage of $V_{T,0} + 6\sigma_{V_T}$ and that caused by a nominal transistor provides a feeling for the impact of transistor variations on delay. This ratio is

$$\frac{D_{6\sigma}}{D_{nominal}} \approx \left(\frac{Vdd - V_{T,0}}{Vdd - V_{T,0} - 6\sigma_{V_T}}\right)^{\alpha}$$

Table D.3 shows the trend of this ratio for the three different types of transistors. The impact is clearly worse for the low standby power and low operating power devices than for the high-performance devices, due to the lower $Vdd - V_T$ for the low power devices.

The delay ratio should not be interpreted as a direct indication of speed loss for the entire system, as they indicate the "worst-case" delay of a single minimal-sized gate. This gate is part of a chain of gates with random delay variations, and several of these gates use larger transistors. These facts moderate the total delay variations somewhat.

It seems that as from $22nm$ on, most of the speed improvement due to the nominal scaling trend $\tau \sim \lambda$ is lost due to the impact of transistor variations. At the $10nm$ node, the gate yield drops belows the 6σ-level (unless subthreshold operation is considered acceptable). A larger Vdd or calibration or redundancy schemes must be introduced to provide acceptable system-level yield.

Transistor variations have little direct impact on the dynamic energy consumption. However, the increasing delay variations can increase the number of glitches. Glitch-free circuit styles hence become more attractive.

Logic at reduced supply voltage

In a $90nm$ memory design, the dynamic energy consumption of decoder and sense amplifiers can be reduced with a factor of 4 by reducing the supply voltage

λ $[nm]$	90	65	45	32	22	16	10
$A_{\Delta V_T}$ $[mV \cdot \mu m]$	3.5	3.0	2.5	2.1	1.7	1.5	1.2
$\sigma_{V_T,2\lambda}$ $[mV]$	19	23	27	33	39	46	58
read operation							
$\sigma_{SNM_{read}}$ $[mV]$	11.7	13.7	16.5	19.6	23.6	27.7	35.0
$\beta = 1$; $\mu = 140mV$	12.0σ	10.2σ	8.5σ	7.2σ	5.9σ	5.1σ	4.0σ
$\beta = 2$; $\mu = 180mV$	15.4σ	13.1σ	10.9σ	9.2σ	7.6σ	6.5σ	5.1σ
$\beta = 3$; $\mu = 200mV$	17.1σ	14.6σ	12.1σ	10.2σ	8.5σ	7.3σ	5.7σ
data retention							
$\sigma_{SNM_{hold}}$ $[mV]$	7.8	9.2	11.0	13.0	15.7	18.5	23.3
$\mu = 380mV$	49σ	42σ	35σ	29σ	24σ	21σ	16σ
Vdd_{min} $[mV]$	213	230	252	277	309	320	400
write operation							
σ_{WTP} $[mV]$	19	23	28	33	39	46	58
$\mu = 380mV - 100mV$	14.4σ	12.2σ	10.2σ	8.6σ	7.1σ	6.1σ	4.8σ

Table D.4: Scaling of the 6T cell, assuming $Vdd = 1V$.

from $1.2V$ to $0.6V$. This increases the total memory delay with a factor of 3.5. In further scaled technology nodes with similar threshold voltage and lower Vdd, this available trade-off range becomes smaller. According to table D.3, no further reduction in Vdd is feasible at the $10nm$ node for the low power technologies, as even the nominal supply voltage does not provide an acceptable yield.

Two approaches might help to allow the use of lower supply voltages. First, transistors with lower threshold voltages can be used when they are combined with efficient power gating strategies to suppress the leakage. Second, the impact of transistor variations can be mitigated with calibration and redundancy. For example, segments of the decoder that do not work at the reduced supply voltage Vdd_{low} can be configured to use the nominal supply voltage. For regular structures such as the decoder, such schemes can probably reduce the feasible Vdd_{low} by 3 to 4 σ_{V_T}, that is $140mV$ to $180mV$ for the $16nm$ node.

Memory cell

Table D.4 summarizes the cell metrics for the different operations. The values for mean and standard deviation of the cell margins are adapted from section 2.5, with updated assumptions for $A_{\Delta V_T}$. In the table, an additional $100mV$ margin on the WTP is enforced to allow for a finite write time and finite impedance of the write drivers. The functional yield of the 6T cell stays above 6σ down to the $22nm$ node. Beyond this node, changes must be introduced. Several options seem sufficient down to $10nm$: the use of different bias voltages for different operation modes or the use of dynamic read stability, a different cell topology

$\lambda\ [nm]$	90	65	45	32	22	16	10
$A_{\Delta V_T}\ [mV \cdot \mu m]$	3.5	3.0	2.5	2.1	1.7	1.5	1.2
$\sigma_{V_T,2\lambda}\ [mV]$	19	23	27	33	39	46	58
Required ΔV_{min} (for transistor mismatch)							
$W = 2\lambda\ [mV]$	165	194	233	277	334	391	495
$W = 8\lambda\ [mV]$	83	97	117	138	167	196	247
Required transistor width (for transistor mismatch)							
$\Delta V_{min} = 100mV$	5.4λ	7.5λ	11λ	15λ	22λ	31λ	49λ
$\Delta V_{min} = 50mV$	22λ	30λ	44λ	61λ	89λ	123λ	196λ
$\Delta V_{min} = 25mV$	87λ	121λ	174λ	245λ	356λ	490λ	784λ
Magnitude of thermal noise (drain-input latch-type SA)							
$W = 8\lambda$							
$\sigma_{V_{offset}}\ [mV]$	1.8	2.2	2.7	3.4	4.3	5.2	7.0
$10 \cdot \sigma_{V_{offset}}\ [mV]$	18	22	27	34	43	52	70
$W = 2\lambda$							
$\sigma_{V_{offset}}\ [mV]$	2.5	3.1	3.9	4.8	6.1	7.4	9.9
$10 \cdot \sigma_{V_{offset}}\ [mV]$	25	31	39	48	61	74	99

Table D.5: Scaling of low-swing data transfers. A 6σ yield target is used for all mismatch-related calculations.

or cell redundancy.

The cell delay suffers in a similar way from variations as the logic circuits. However, there are two factors that make the problem worse. First, in memory architectures in which the cell delay is responsible for a large part of the memory delay, the delay variations are not averaged out. Second, because there are so many cells and these cells cannot be power gated in typical applications, the cell transistors must use high threshold voltages. The increase in delay can be mitigated by using readbuffers or local sense amplifiers, or weak cells can be avoided with redundancy.

Thermal noise was ignored in this exercise, even though it might become important for memory cells. The values for the sense amplifier offset voltage due to thermal noise for $W = 2\lambda$ in table D.5 provide a first indication of the magnitude of the problem.

Low-swing data transfers

The scaling of low-swing data transfers is somewhat problematic. Table D.5 provides numerical examples.

The delay of a low-swing data transfer is given by $\frac{C \cdot V_{swing}}{I_{read}}$, the dynamic energy by $C \cdot V_{swing} \cdot Vdd$. The minimal required differential input voltage ΔV_{min} of the sense amplifier is also the minimal required V_{swing} for a memory with complementary bit lines.

If the sense amplifier is scaled down along with the other circuits, ΔV_{min} increases as $\sim \frac{A_{\Delta V_T}}{\sqrt{W \cdot L}} \sim \frac{1}{\sqrt{\lambda}}$. For the dynamic energy consumption, such scaling would result in

$$E_{bit\ line} = C \cdot V_{swing} \cdot Vdd \sim \lambda^{1.25} \cdot \frac{1}{\sqrt{\lambda}} \cdot \lambda^{0.25} \sim \lambda$$

Because the other dynamic energy components reduce faster (as $E \sim \lambda^{1.75}$), the importance of the bit line energy consumption would increase quickly. For the delay, such scaling results in

$$D_{bit\ line} = \frac{C \cdot V_{swing}}{I_{read}} \sim \frac{\lambda^{1.25} \cdot \frac{1}{\sqrt{\lambda}}}{\sqrt{\lambda}} \sim \sqrt[4]{\lambda}$$

Because the other delay components reduce faster (as $\sim \lambda$), the importance of the bit line read delay would quickly increase.

On the other hand, one could try to keep ΔV_{min} constant while scaling, for example at $100mV$. With this choice, the bit line energy and delay reduce only slightly less quickly with scaling than that of the other memory components. However, with a traditional sense amplifier implementation, the importance of the sense amplifier energy consumption would increase very fast. The use of sense amplifier calibration hence becomes very important with scaling.

Table D.5 also shows another challenge for the sense amplifiers: thermal noise. As discussed in section 4.2.3, thermal noise is proportional to $\sim \frac{1}{\sqrt{C_0}} \sim \frac{1}{\sqrt{\lambda^{1.25}}}$, with C_0 the capacitance on the sensitive internal nodes of the sense amplifier. The $\sigma_{V_{offset}}$ due to thermal noise increases quickly with scaling. The values in the table are provided for the drain-input latch-type sense amplifier which has $A_{pre} = 1$. As discussed in section 4.6.7, $\sigma_{V_{offset}}$ of sense amplifiers with pre-amplification before positive feedback kicks in is a factor A_{pre} smaller. It is reasonably easy to obtain $A_{pre} = 2.6$ with a pulsed current source SA, but it will require other sense amplifier designs to obtain a significantly larger pre-amplification.

If a calibrated SA with pre-amplification is combined with a DC-DC convertor for the generation of the precharge voltage for the global bit lines, the data transfers should not be the limiting factor for the further performance scaling of memories down to the $10nm$ node.

Summary

The most important challenges for SRAM scaling are related to the low-swing data transfers and to the memory cells. The techniques that are discussed in this text seem to suffice to solve both challenges down to $10nm$.

With scaling, the nominal energy and delay of the memory improve more or less as follows:

- $E_{dynamic} \sim C \cdot Vdd^2 \sim \lambda^{1.75}$
- $P_{leakage} \sim I_{leakage} \cdot Vdd \sim \lambda^{1.25}$
- $Delay \sim \frac{C \cdot Vdd}{I_{dsat}} \sim \lambda$

Transistor variations will reduce the actual improvement in memory delay and energy, but at least down to $22nm$, clear advantages are expected. The energy consumption reduces down to $10nm$.

Only stochastic intra-die variations are considered in this analysis. Die-to-die variations, systematic variations and the impact of aging and temperature are ignored. Thermal noise is only considered for the sense amplifiers. Including these effects in the analysis would quite likely result in a significantly less bright picture.

Appendix E

List of Publications and Biography

Conference papers

[CP-1] **Stefan Cosemans**, Wim Dehaene, Francky Catthoor, "A Low Power Embedded SRAM for Wireless Applications", in *Proceedings of 32nd European Solid-State Circuits Conference (ESSCIRC)*, pp. 291-294, September 2006

[CP-2] **Stefan Cosemans**, Wim Dehaene, Francky Catthoor, "A 3.6pJ/access 480MHz, 128Kbit on-Chip SRAM with 850MHz boost mode in 90nm CMOS with tunable sense amplifiers to cope with variability", in *Proceedings of 34nd European Solid-State Circuits Conference (ESSCIRC)*, pp. 278-281, September 2008

[CP-3] Wim Dehaene, **Stefan Cosemans**, Anselme Vignon, Francky Catthoor, Peter Geens, "Embedded SRAM design in deep deep submicron technologies", in *Proceedings of 33rd European Solid-State Circuits Conference (ESSCIRC)*, pp. 384-391, September 2007

[CP-4] Staf Verhaegen, **Stefan Cosemans**, Mircea Dusa, Pol Marchal, Axel Nackaerts, Geert Vandenberghe, Wim Dehaene, "Litho variations and their impact on the electrical yield of a 32nm node 6T SRAM cell", in *SPIE Proceedings, Design for Manufacturability through Design-Process Integration II*, VOL. 6925, 69250R (2008)

[CP-5] Anselme Vignon, **Stefan Cosemans**, Pol Marchal, Marco Facchini, Wim Dehaene, "A novel DRAM architecture as a low leakage alternative for SRAM caches in a 3D interconnect context", accepted for publication in *Proceedings of Design, Automation and Test in Europe, DATE '09*, 2009

Journal papers

[JP-1] **Stefan Cosemans**, Wim Dehaene, Francky Catthoor, "A low power embedded SRAM for wireless applications", in *IEEE Journal of Solid-State Circuits*, vol. 42, no. 7, pp. 1607-1617, July, 2007

[JP-2] **Stefan Cosemans**, Wim Dehaene, Francky Catthoor, "A 3.6pJ/Access 480MHz, 128Kbit on-Chip SRAM with 850MHz Boost Mode in 90nm CMOS with Tunable Sense Amplifiers", accepted for publication in *IEEE Journal of Solid-State Circuits*, vol. 44, no. 7, July 2009

Biography

Stefan Cosemans was born in Mol, Belgium, in 1981. He received the M. Sc. degree in electrical engineering from the Katholieke Universiteit Leuven (K.U.Leuven), Heverlee, Belgium, in 2004. The subject of his M.S. thesis was the design of an operating system extension for runtime reconfigurable platforms.

From October 2004 till now he has been a research assistant at the ESAT-MICAS laboratory of the Katholieke Universiteit Leuven. He has been working towards the Ph.D. degree on the variability-aware design of low power embedded memories.

Bibliography

[Amr94] B.S. Amrutur and M. Horowitz, "Techniques to reduce power in fast wide memories [CMOS SRAMs]", In *Digest of Technical Papers IEEE Symposium Low Power Electronics*, pages 92–93, 1994.

[Amr98] B.S. Amrutur and M.A. Horowitz, "A replica technique for wordline and sense control in low-power SRAM's", *IEEE Journal of Solid-State Circuits (JSSC)*, 33(8):1208–1219, 1998.

[Amr00] B.S. Amrutur and M.A. Horowitz, "Speed and power scaling of SRAM's", *IEEE Journal of Solid-State Circuits (JSSC)*, 35(2):175–185, 2000.

[AR08] M.H. Abu-Rahma, M. Anis, and Sei Seung Yoon, "A robust single supply voltage SRAM read assist technique using selective precharge", In *Proceedings European Solid-State Circuits Conference (ESSCIRC)*, pages 234–237, 2008.

[Asa05] T. Asano, J. Silberman, S.H. Dhong, O. Takahashi, M. White, S. Cottier, T. Nakazato, A. Kawasumi, and H. Yoshihara, "Low-power design approach of 11FO4 256-Kbyte embedded SRAM for the synergistic processor element of a Cell processor", *IEEE Micro*, 25(5):30–38, 2005.

[Bam04] Mandeep Bamal, Evelyn Grossar, Michele Stucchi, and Karen Maex, "Interconnect width selection for deep submicron designs using the table lookup method", In *Proceedings International Workshop on System Level Interconnect Prediction (SLIP)*, pages 41–44, New York, NY, USA, 2004. ACM.

[Bar08] S. Barasinski, L. Camus, and S. Clerc, "A 45nm single power supply SRAM supporting low voltage operation down to 0.6V", In *Proceedings European Solid-State Circuits Conference (ESSCIRC)*, pages 502–505, 2008.

[Bas03] T. Basten, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits, 2/E*, Prentice Hall, 2003.

[Bha01] A.J. Bhavnagarwala, Xinghai Tang, and J.D. Meindl, "The impact of intrinsic device fluctuations on CMOS SRAM cell stability", *IEEE Journal of Solid-State Circuits (JSSC)*, 36(4):658–665, Apr 2001.

[Bha03] A.J. Bhavnagarwala, S.V. Kosonocky, M. Immediato, D. Knebel, and A.-M. Haen, "A pico-joule class, 1 GHz, 32 KByte64 b DSP SRAM with self reverse bias", In *Proceedings IEEE Symposium on VLSI Circuits*, pages 251–252, 2003.

[Bha08] A.J. Bhavnagarwala, S. Kosonocky, C. Radens, Yuen Chan, K. Stawiasz, U. Srinivasan, S.P. Kowalczyk, and M.M. Ziegler, "A Sub-600-mV, Fluctuation Tolerant 65-nm CMOS SRAM Array With Dynamic Cell Biasing", *IEEE Journal of Solid-State Circuits (JSSC)*, 43(4):946–955, 2008.

[Cal06] B.H. Calhoun and A.P. Chandrakasan, "Static noise margin variation for sub-threshold SRAM in 65-nm CMOS", *IEEE Journal of Solid-State Circuits (JSSC)*, 41(7):1673–1679, 2006.

[Cal07] Benton Highsmith Calhoun and Anantha P. Chandrakasan, "A 256-kb 65-nm Sub-threshold SRAM Design for Ultra-Low-Voltage Operation", *IEEE Journal of Solid-State Circuits (JSSC)*, 42(3):680–688, 2007.

[Car04] I. Carlson, S. Andersson, S. Natarajan, and A. Alvandpour, "A high density, low leakage, 5T SRAM for embedded caches", In *Proceedings European Solid-State Circuits Conference (ESSCIRC)*, pages 215–218, 2004.

[Cha05a] L. Chang, D.M. Fried, J. Hergenrother, J.W. Sleight, R.H. Dennard, R.K. Montoye, L. Sekaric, S.J. McNab, A.W. Topol, C.D. Adams, and et al., "Stable SRAM cell design for the 32 nm node and beyond", In *Digest of Technical Papers IEEE International Symposium on VLSI Technology*, pages 128–129, 2005.

[Cha05b] Yen-Jen Chang and Feipei Lai, "Dynamic zero-sensitivity scheme for low-power cache memories", *IEEE Micro*, 25(4):20–32, 2005.

[Che07] Kuan-Lun Cheng, C.C. Wu, Y.P. Wang, D.W. Lin, C.M. Chu, Y.Y. Tarng, S.Y. Lu, S.J. Yang, M.H. Hsieh, C.M. Liu, and et al., "A Highly Scaled, High Performance 45nm Bulk Logic CMOS Technology with 0.242 um2 SRAM Cell", In *Proceedings IEEE International Electron Devices Meeting (IEDM)*, pages 243–246, 2007.

[Cla01] L.T. Clark, E.J. Hoffman, J. Miller, M. Biyani, Luyun Liao, S. Strazdus, M. Morrow, K.E. Velarde, and M.A. Yarch, "An embedded 32-b microprocessor core for low-power and high-performance applications", *IEEE Journal of Solid-State Circuits (JSSC)*, 36(11):1599–1608, Nov 2001.

[Cos06] Stefan Cosemans, Wim Dehaene, and Francky Catthoor, "A Low Power Embedded SRAM for Wireless Applications", In *Proceedings European Solid-State Circuits Conference (ESSCIRC)*, pages 291–294, 2006.

[Cos07] S. Cosemans, W. Dehaene, and F. Catthoor, "A Low-Power Embedded SRAM for Wireless Applications", *IEEE Journal of Solid-State Circuits (JSSC)*, 42(7):1607–1617, 2007.

[Cos08] Stefan Cosemans, Wim Dehaene, and Francky Catthoor, "A 3.6pJ/access 480MHz, 128Kbit on-Chip SRAM with 850MHz boost mode in 90nm CMOS with tunable sense amplifiers to cope with variability", In *Proceedings European Solid-State Circuits Conference (ESSCIRC)*, pages 278–281, 2008.

[Cos09] S. Cosemans, W. Dehaene, and F. Catthoor, "A 3.6pJ/Access 480MHz, 128Kbit on-Chip SRAM with 850MHz Boost Mode in 90nm CMOS with Tunable Sense Amplifiers", *accepted for publication in IEEE Journal of Solid-State Circuits*, 44(7), 2009.

[Dec05] G. Declerck, "A look into the future of nanoelectronics", In *Digest of Technical Papers IEEE International Symposium on VLSI Technology*, pages 6–10, June 2005.

[DN07] M. De Nil, L. Yseboodt, F. Bouwens, J. Hulzink, M. Berekovic, J. Huisken, and J. van Meerbergen, "Ultra Low Power ASIP Design for Wireless Sensor Nodes", In *Proceedings of the IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pages 1352–1355, Dec. 2007.

[Doo08] T.S. Doorn, E.J.W. ter Maten, J.A. Croon, A. Di Bucchianico, and O. Wittich, "Importance sampling Monte Carlo simulations for accurate estimation of SRAM yield", In *Proceedings European Solid-State Circuits Conference (ESSCIRC)*, pages 230–233, 2008.

[Dor07] J. Dorsey, S. Searles, M. Ciraula, S. Johnson, N. Bujanos, D. Wu, M. Braganza, S. Meyers, E. Fang, and R. Kumar, "An Integrated Quad-Core Opteron Processor", In *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC)*, pages 102–103, 2007.

[eld07] *Eldo User's Manual. Software Version 6.9 1 Release AMS 2007.1*, Mentor Graphics, 2007.

[Gee05] Peter Geens and Wim Dehaene, "A small granular controlled leakage reduction system for SRAMs", *Solid-State Electronics*, 49(11):1776–1782, Nov 2005.

[Gee08] Peter Geens and Wim Dehaene, "A dual port dual width 90nm SRAM with guaranteed data retention at minimal standby supply voltage", In *Proceedings European Solid-State Circuits Conference (ESSCIRC)*, pages 290–293, 2008.

[Gee09] Peter Geens, *Active Supply Control in Static Random Access Memories*, PhD thesis, Katholieke Universiteit Leuven (KUL), 2009.

[Gep04] L. Geppert, "A static RAM says goodbye to data errors [radiation induced soft errors]", *IEEE Spectrum*, 41(2):16–17, 2004.

[Gie05] G. Gielen and W. Dehaene, "Analog and digital circuit design in 65 nm CMOS: end of the road?", In *Proceedings of Design, Automation and Test in Europe (DATE)*, pages 37–42 Vol. 1, March 2005.

[Gro06] E. Grossar, M. Stucchi, K. Maex, and W. Dehaene, "Read Stability and Write-Ability Analysis of SRAM Cells for Nanometer Technologies", *IEEE Journal of Solid-State Circuits (JSSC)*, 41(11):2577–2588, 2006.

[Haj98] A. Hajimiri and R. Heald, "Design issues in cross-coupled inverter sense amplifier", In *Proceedings IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 2, pages 149–152 vol.2, 1998.

[Ham09] F. Hamzaoglu, K. Zhang, Y. Wang, H. J. Ahn, U. Bhattacharya, Z. Chen, Y.-G. Ng, A. Pavlov, K. Smits, and M. Bohr, "A 3.8 GHz 153 Mb SRAM Design With Dynamic Stability Enhancement and Leakage Reduction in 45 nm High-k Metal Gate CMOS Technology", *IEEE Journal of Solid-State Circuits (JSSC)*, 44(1):148–154, 2009.

[Hau93] J.R. Hauser, "Noise margin criteria for digital logic circuits", *IEEE Transactions on Education*, 36(4):363–368, 1993.

[Haz03] P. Hazucha, T. Karnik, S. Walstra, B. Bloechel, J. Tschanz, J. Maiz, K. Soumyanath, G. Dermer, S. Narendra, V. De, and et al., "Measurements and analysis of SER tolerant latch in a 90 nm dual-Vt CMOS process", In *Proceedings of IEEE Custom Integrated Circuits Conference (CICC)*, pages 617–620, 2003.

[Hei08] D.F. Heidel, K.P. Rodbell, E.H. Cannon, C. Calabral, M.S. Gordon, P. Oldiges, and H.H.K. Tang, "Alpha-particle-induced upsets in advanced CMOS circuits and technology", *IBM Journal of Research and Development*, 52(3):225–232, 2008.

[Hil68] C.F. Hill, "Noise margin and noise immunity of logic circuits", *Microelectronics*, 1:16–21, 1968.

[Hir09] O. Hirabayashi and e.a., "A Process-Variation-Tolerant Dual-Power-Supply SRAM with a 0.179um2 Cell in 40nm CMOS using Level-Programmable Wordline Driver", In *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC)*, pages 458–459, 2009.

[Hou08] R.M. Houle, "Simple Statistical Analysis Techniques to Determine Optimum Sense Amp Set Times", *IEEE Journal of Solid-State Circuits (JSSC)*, 43(8):1816–1825, 2008.

[Ish98] M. Ishida, T. Kawakami, A. Tsuji, N. Kawamoto, M. Motoyoshi, and N. Ouchi, "A novel 6T-SRAM cell technology designed with rectangular patterns scalable beyond 0.18 m generation and desirable for ultra high speed operation", In *Proceedings IEEE International Electron Devices Meeting (IEDM)*, pages 201–204, 1998.

[Ito95] K. Itoh, K. Sasaki, and Y. Nakagome, "Trends in low-power RAM circuit technologies", *Proceedings of the IEEE*, 83(4):524–543, 1995.

[Ito09] K Itoh, "Adaptive Circuits for the 0.5-V Nanoscale CMOS Era", In *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC)*, pages 14–20, 2009.

[Iye05] S Iyer, S, J.E. Barth, P.C. Parries, J.P. Norum, J.P Rice, L.R. Logan, and D. Hoyniak, "Embedded DRAM: Technology platform for the Blue Gene/L chip", *IBM Journal of Research and Development*, 49(2/3), May 2005.

[Jos08] R. Joshi, R. Houle, D. Rodko, P. Patel, W. Huott, R. Franch, Y. Chan, D. Plass, S. Wilson, S. Wu, and et al., "A high performance 2.4 Mb L1 and L2 cache compatible 45nm SRAM with yield improvement capabilities", In *Proceedings IEEE Symposium on VLSI Circuits*, pages 208–209, 2008.

[Jun03] Soon-Moon Jung, Hoon Lim, Wonseok Cho, Hoosung Cho, Hatae Hong, Jaehun Jeong, Sugwoo Jung, Hanbyung Park, Byoungkeun Son, Youngchul Jang, and et al., "Soft error immune 0.46 m2 SRAM cell with MIM node capacitor by 65 nm CMOS technology for ultra high speed SRAM", In *Proceedings IEEE International Electron Devices Meeting (IEDM)*, pages 11.4.1–11.4.4, 2003.

[Kan06] R. Kanj, R. Joshi, and S. Nassif, "Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events", In *Proceedings of the Design Automation Conference (DAC)*, pages 69–72, 2006.

[Kar98] A. Karandikar and K.K. Parhi, "Low power SRAM design using hierarchical divided bit-line approach", In *Proceedings International Conference on Computer Design: VLSI in Computers and Processors (ICCD)*, pages 82–88, 1998.

[Kar04] T. Karnik and P. Hazucha, "Characterization of soft errors caused by single event upsets in CMOS processes", *IEEE Transactions on Dependable and Secure Computing*, 1(2):128–143, 2004.

[Kaw08a] A. Kawasumi, T. Yabe, Y. Takeyama, O. Hirabayashi, K. Kushida, A. Tohata, T. Sasaki, A. Katayama, G. Fukano, Y. Fujimura, and e.a., "A Single-Power-Supply 0.7V 1GHz 45nm SRAM with An Asymmetrical Unit-β-ratio Memory Cell", In *Digest of Technical*

Papers IEEE International Solid-State Circuits Conference (ISSCC), pages 382–622, 2008.

[Kaw08b] A. Kawasumi, T. Yabe, Y. Takeyama, O. Hirabayashi, K. Kushida, A. Tohata, T. Sasaki, A. Katayama, G. Fukano, Y. Fujimura, and et al., "A Single-Power-Supply 0.7V 1GHz 45nm SRAM with An Asymmetrical Unit-β-ratio Memory Cell", In *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC)*, pages 382–622, 2008.

[Kha02] M. Khare, S.H. Ku, R.A. Donaton, S. Greco, C. Brodsky, X. Chen, A. Chou, R. DellaGuardia, S. Deshpande, B. Doris, and et al., "A high performance 90nm SOI technology with 0.992 μm^2 6T-SRAM cell", In *Proceedings IEEE International Electron Devices Meeting (IEDM)*, pages 407–410, 2002.

[Khe06] M. Khellah, Y. Ye, N. Kim, D. Somasekhar, G. Pandya, A. Farhang, K. Zhang, C. Webb, and V. De, "Wordline and Bitline Pulsing Schemes for Improving SRAM Cell Stability in Low-Vcc 65nm CMOS Designs", In *Proceedings IEEE Symposium on VLSI Circuits*, pages 9–10, 2006.

[Khe07] M. Khellah, D.E. Khalil, D. Somasekhar, Y. Ismail, T. Karnik, and V. De, "Effect of Power Supply Noise on SRAM Dynamic Stability", In *Proceedings IEEE Symposium on VLSI Circuits*, pages 76–77, 2007.

[Khe08] M.M. Khellah, A. Keshavarzi, D. Somasekhar, T. Karnik, and V. De, "Read and write circuit assist techniques for improving Vccmin of dense 6T SRAM cell", In *Proceedings IEEE International Conference on Integrated Circuit Design and Technology and Tutorial (ICICDT)*, pages 185–188, 2008.

[Kim96] Gyudong Kim, Min-Kyu Kim, Byoung-Soo Chang, and Wonchan Kim, "A low-voltage, low-power CMOS delay element", *IEEE Journal of Solid-State Circuits (JSSC)*, 31(7):966–971, 1996.

[Kim05] C.H. Kim, Jae-Joon Kim, S. Mukhopadhyay, and K. Roy, "A forward body-biased low-leakage SRAM cache: device, circuit and architecture considerations", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(3):349–357, 2005.

[Kim08] Keejong Kim, H. Mahmoodi, and K. Roy, "A Low-Power SRAM Using Bit-Line Charge-Recycling", *IEEE Journal of Solid-State Circuits (JSSC)*, 43(2):446–459, 2008.

[Kob92] T. Kobayashi, K. Nogami, T. Shirotori, Y. Fujimoto, and O. Watanabe, "A current-mode latch sense amplifier and a static power saving input buffer for low-power architecture", In *Proceedings IEEE Symposium on VLSI Circuits*, pages 28–29, 1992.

[Kra89] R. Kraus, "Analysis and reduction of sense-amplifier offset", *IEEE Journal of Solid-State Circuits (JSSC)*, 24(4):1028–1033, 1989.

[Kru08] D. Krueger, E. Francom, and J. Langsdorf, "Circuit Design for Voltage Scaling and SER Immunity on a Quad-Core Itanium Processor", In *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC)*, pages 94–95, 2008.

[Kul07] J.P. Kulkarni, K. Kim, and K. Roy, "A 160 mV Robust Schmitt Trigger Based Subthreshold SRAM", *IEEE Journal of Solid-State Circuits (JSSC)*, 42(10):2303–2313, 2007.

[Kul08] J.P. Kulkarni, K. Kim, S.P. Park, and K. Roy, "Process variation tolerant SRAM array for ultra low voltage applications", In *Proceedings of the Design Automation Conference (DAC)*, pages 108–113, 2008.

[Kus09] K. Kushida, A. Suzuki, G. Fukano, A. Kawasumi, O. Hirabayashi, Y. Takeyama, T. Sasaki, A. Katayama, Y. Fujimura, and T. Yabe, "A 0.7 V Single-Supply SRAM With 0.495 μm^2 Cell in 65 nm Technology Utilizing Self-Write-Back Sense Amplifier and Cascaded Bit Line Scheme", *IEEE Journal of Solid-State Circuits (JSSC)*, 44(4):1192–1198, 2009.

[Kwo08] J. Kwong, Y. Ramadass, N. Verma, M. Koesler, K. Huber, H. Moormann, and A. Chandrakasan, "A 65nm Sub-Vt Microcontroller with Integrated SRAM and Switched-Capacitor DC-DC Converter", In *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC)*, pages 318–616, 2008.

[Lai09] Y.-C. Lai and S.-Y. Huang, "Robust SRAM Design via BIST-Assisted Timing-Tracking (BATT)", *IEEE Journal of Solid-State Circuits (JSSC)*, 44(2):642–649, 2009.

[Lam05] A. Lambrechts, P. Raghavan, A. Leroy, G. Talavera, T.V. Aa, M. Jayapala, F. Catthoor, D. Verkest, G. Deconinck, H. Corporaal, F. Robert, and J. Carrabina, "Power breakdown analysis for a heterogeneous NoC platform running a video application", In *Proceedings IEEE International Conference on Application-Specific Systems and Architecture Processors (ASAP)*, pages 179–184, July 2005.

[Lee08] Yoonmyung Lee, Mingoo Seok, Scott Hanson, David Blaauw, and Dennis Sylvester, "Standby power reduction techniques for ultra-low power processors", In *Proceedings European Solid-State Circuits Conference (ESSCIRC)*, pages 186–189, 2008.

[Loh83] J. Lohstroh, E. Seevinck, and J. de Groot, "Worst-case static noise margin criteria for logic circuits and their mathematical equivalence", *IEEE Journal of Solid-State Circuits (JSSC)*, 18(6):803–807, 1983.

[Mai98] K.W. Mai, T. Mori, B.S. Amrutur, R. Ho, B. Wilburn, M.A. Horowitz, I. Fukushi, T. Izawa, and S. Mitarai, "Low-power SRAM design using half-swing pulse-mode techniques", *IEEE Journal of Solid-State Circuits (JSSC)*, 33(11):1659–1671, 1998.

[Mar09] E. Maricau and G. Gielen, "Efficient Reliability Simulation of Analog ICs Including Variability and Time-varying Stress", In *Proceedings of Design, Automation and Test in Europe (DATE)*, 2009.

[Mas02] K. Masselos, F. Catthoor, C.E. Goutis, and H. Deman, "A systematic methodology for the application of data transfer and storage optimizing code transformations for power consumption and execution time reduction in realizations of multimedia algorithms on programmable processors", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 10(4):515–518, Aug 2002.

[McC05] T. McConaghy, T. Eeckelaert, and G. Gielen, "CAFFEINE: template-free symbolic model generation of analog circuits via canonical form functions and genetic programming", In *Proceedings European Solid-State Circuits Conference (ESSCIRC)*, pages 243–246, 2005.

[McC06] T. McConaghy and G. Gielen, "Double-Strength CAFFEINE: Fast Template-Free Symbolic Modeling of Analog Circuits via Implicit Canonical Form Functions and Explicit Introns", In *Proceedings of Design, Automation and Test in Europe (DATE)*, volume 1, pages 1–6, 2006.

[McC09] Trent McConaghy, *Variation-Aware Structural Synthesis and Knowledge Extraction of Analog Circuits*, PhD thesis, Katholieke Universiteit Leuven (KUL), 2009.

[Mis07] K. Mistry, C. Allen, C. Auth, B. Beattie, D. Bergstrom, M. Bost, M. Brazier, M. Buehler, A. Cappellani, R. Chau, and et al., "A 45nm Logic Technology with High-k+Metal Gate Transistors, Strained Silicon, 9 Cu Interconnect Layers, 193nm Dry Patterning, and 100% Pb-free Packaging", In *Proceedings IEEE International Electron Devices Meeting (IEDM)*, pages 247–250, 2007.

[Mon03] R. Montoye, W. Belluomini, H. Ngo, C. McDowell, J. Sawada, T. Nguyen, B. Veraa, J. Wagoner, and M. Lee, "A double precision floating point multiply", In *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC)*, pages 336–337 vol.1, 2003.

[Mor98] T. Mori, B. Amrutur, K. Mai, M. Horowitz, I. Fukushi, T. Izawa, and S. Mitarai, "A 1 V 0.9 mW at 100 MHz 2K x 16b SRAM utilizing a half-swing pulsed-decoder and write-bus architecture in 0.25 um

dual-Vt CMOS", In *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC)*, pages 354–355, 463, 1998.

[Muk08] Shubu Mukherjee, *Architecture Design for Soft Errors*, Morgan Kaufmann, 2008.

[Nam98] H. Nambu, K. Kanetani, K. Yamasaki, K. Higeta, M. Usami, Y. Fujimura, K. Ando, T. Kusunoki, K. Yamaguchi, and N. Homma, "A 1.8-ns access, 550-MHz, 4.5-Mb CMOS SRAM", *IEEE Journal of Solid-State Circuits (JSSC)*, 33(11):1650–1658, 1998.

[Nar06] S. Narasimha, K. Onishi, H.M. Nayfeh, A. Waite, M. Weybright, J. Johnson, C. Fonseca, D. Corliss, C. Robinson, M. Crouse, and et al., "High Performance 45-nm SOI Technology with Enhanced Strain, Porous Low-κ BEOL, and Immersion Lithography", In *Proceedings IEEE International Electron Devices Meeting (IEDM)*, pages 1–4, 2006.

[Nii04] K. Nii, Y. Tsukamoto, T. Yoshizawa, S. Imaoka, Y. Yamagami, T. Suzuki, A. Shibayama, H. Makino, and S. Iwade, "A 90-nm low-power 32-kB embedded SRAM with gate leakage suppression circuit for mobile applications", *IEEE Journal of Solid-State Circuits (JSSC)*, 39(4):684–693, 2004.

[Nii08] K. Nii, M. Yabuuchi, Y. Tsukamoto, S. Ohbayashi, Y. Oda, K. Usui, T. Kawamura, N. Tsuboi, T. Iwasaki, K. Hashimoto, and et al., "A 45-nm single-port and dual-port SRAM family with robust read/write stabilizing circuitry under DVFS environment", In *Proceedings IEEE Symposium on VLSI Circuits*, pages 212–213, 2008.

[Nik00] B. Nikolic, V.G. Oklobdzija, V. Stojanovic, Wenyan Jia, James Kar-Shing Chiu, and M. Ming-Tak Leung, "Improved sense-amplifier-based flip-flop: design and measurements", *IEEE Journal of Solid-State Circuits (JSSC)*, 35(6):876–884, 2000.

[NIS06] *NIST/SEMATECH e-Handbook of Statistical Methods, http://www.itl.nist.gov/div898/handbook/*, 2006.

[Nuz08] P. Nuzzo, F. De Bernardinis, P. Terreni, and G. Van der Plas, "Noise Analysis of Regenerative Comparators for Reconfigurable ADC Architectures", *IEEE Transactions on Circuits and Systems I: Regular Papers*, 55(6):1441–1454, 2008.

[Ohb07] S. Ohbayashi, M. Yabuuchi, K. Nii, Y. Tsukamoto, S. Imaoka, Y. Oda, T. Yoshihara, M. Igarashi, M. Takeuchi, H. Kawashima, and et al., "A 65-nm SoC Embedded 6T-SRAM Designed for Manufacturability With Read and Write Operation Stabilizing Circuits", *IEEE Journal of Solid-State Circuits (JSSC)*, 42(4):820–829, 2007.

[Opr97] I.E. Opris, "Noise estimation in strobed comparators", *Electronics Letters*, 33(15):1273–1274, 1997.

[Osa01] K. Osada, Jinuk Luke Shin, M. Khan, Y. Liou, K. Wang, K. Shoji, K. Kuroda, S. Ikeda, and K. Ishibashi, "Universal-Vdd 0.65-2.0-V 32-kB cache using a voltage-adapted timing-generation scheme and a lithographically symmetrical cell", *IEEE Journal of Solid-State Circuits (JSSC)*, 36(11):1738–1744, 2001.

[Pap05] A. Papanikolaou, F. Lobmaier, H. Wang, M. Miranda, and F. Catthoor, "A system-level methodology for fully compensating process variability impact of memory organizations in periodic applications", In *Proceedings of the 3rd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, pages 117–122. ACM, 2005.

[Pig01] C. Piguet, J.-M. Masgonty, S. Cserveny, C. Arm, and P.-D. Pfister, "Low-power low-voltage library cells and memories", In *Proceedings of the IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, volume 3, pages 1521–1524 vol.3, 2001.

[Pil07] H. Pilo, C. Barwin, G. Braceras, C. Browning, S. Lamphier, and F. Towler, "An SRAM Design in 65-nm Technology Node Featuring Read and Write-Assist Circuits to Expand Operating Voltage", *IEEE Journal of Solid-State Circuits (JSSC)*, 42(4):813–819, 2007.

[Pil08] J. Pille, C. Adams, T. Christensen, S.R. Cottier, S. Ehrenreich, F. Kono, D. Nelson, O. Takahashi, S. Tokito, O. Torreiter, and et al., "Implementation of the Cell Broadband Engine in 65 nm SOI Technology Featuring Dual Power Supply SRAM Arrays Supporting 6 GHz at 1.3 V", *IEEE Journal of Solid-State Circuits (JSSC)*, 43(1):163–171, 2008.

[Rab03] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits, 2/E*, Prentice Hall, 2003.

[Ram09] V. Ramadurai, H. Pilo, J. Andersen, G. Braceras, J. Gabric, D. Geise, S. Lamphier, and Yue Tan, "An 8 Mb SRAM in 45 nm SOI Featuring a Two-Stage Sensing Scheme and Dynamic Power Management", *IEEE Journal of Solid-State Circuits (JSSC)*, 44(1):155–162, 2009.

[Rya08] J.F. Ryan and B.H. Calhoun, "Minimizing Offset for Latching Voltage-Mode Sense Amplifiers for Sub-Threshold Operation", In *Proceedings International Symposium on Quality Electronic Design (ISQED)*, pages 127–132, 2008.

[Sak93] T. Sakurai, "High-speed Circuit Design With Scaled-down MOSFET's and Low Supply Voltage", In *Proceedings IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1487–1490, 1993.

[Sal05] F.R. Saliba, H. Kawaguchi, and T. Sakurai, "Experimental verification of row-by-row variable VDD scheme reducing 95% active leakage power of SRAM's", In *Proceedings IEEE Symposium on VLSI Circuits*, pages 162–165, 2005.

[San06] W. Sansen, *Analog Design Essentials*, Springer Verlag, 2006.

[Sar91] R. Sarpeshkar, J.L. Wyatt, N.C. Lu, and P.D. Gerber, "Mismatch sensitivity of a simultaneously latched CMOS sense amplifier", *IEEE Journal of Solid-State Circuits (JSSC)*, 26(10):1413–1422, 1991.

[Sar93] R. Sarpeshkar, T. Delbruck, and C.A. Mead, "White noise in MOS transistors and resistors", *IEEE Circuits and Devices Magazine*, 9(6):23–29, 1993.

[Sat99] H. Sato, T. Wada, S. Ohbayashi, K. Kozaru, Y. Okamoto, Y. Higashide, T. Shimizu, Y. Maki, R. Morimoto, H. Otoi, and et al., "A 500-MHz pipelined burst SRAM with improved SER immunity", *IEEE Journal of Solid-State Circuits (JSSC)*, 34(11):1571–1579, 1999.

[Sch07] D. Schinkel, E. Mensink, E. Kiumperink, E. van Tuijl, and B. Nauta, "A Double-Tail Latch-Type Voltage Sense Amplifier with 18ps Setup+Hold Time", In *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC)*, pages 314–605, 2007.

[See87] E. Seevinck, F.J. List, and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells", *IEEE Journal of Solid-State Circuits (JSSC)*, 22(5):748–754, 1987.

[See91] E. Seevinck, P.J. van Beers, and H. Ontrop, "Current-mode techniques for high-speed VLSI circuits with application to current sense amplifier for CMOS SRAM's", *IEEE Journal of Solid-State Circuits (JSSC)*, 26(4):525–536, 1991.

[Seo08] Mingoo Seok, S. Hanson, Yu-Shiang Lin, Zhiyoong Foo, Daeyeon Kim, Yoonmyung Lee, Nurrachman Liu, D. Sylvester, and D. Blaauw, "The Phoenix Processor: A 30pW platform for sensor applications", In *Proceedings IEEE Symposium on VLSI Circuits*, pages 188–189, June 2008.

[Sha09] M. Sharifkhani and M. Sachdev, "SRAM Cell Stability: A Dynamic Perspective", *IEEE Journal of Solid-State Circuits (JSSC)*, 44(2):609–619, 2009.

[Sil98] J. Silberman, N. Aoki, D. Boerstler, J.L. Burns, Sang Dhong, A. Essbaum, U. Ghoshal, D. Heidel, P. Hofstee, Kyung Tek Lee, and et al., "A 1.0-GHz single-issue 64-bit powerPC integer processor", *IEEE Journal of Solid-State Circuits (JSSC)*, 33(11):1600–1608, 1998.

[Sin04] R. Singh and N. Bhat, "An Offset Compensation Technique for Latch Type Sense Amplifiers in High-speed Low-power SRAMs", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12(6):652–657, 2004.

[Sin07] A. Singhee and R.A. Rutenbar, "Statistical Blockade: A Novel Method for Very Fast Monte Carlo Simulation of Rare Circuit Events, and its Application", In *Proceedings of Design, Automation and Test in Europe (DATE)*, pages 1–6, 2007.

[Sin08a] M.E. Sinangil, N. Verma, and A.P. Chandrakasan, "A reconfigurable 65nm SRAM achieving voltage scalability from 0.25-1.2V and performance scalability from 20kHz-200MHz", In *Proceedings European Solid-State Circuits Conference (ESSCIRC)*, pages 282–285, 2008.

[Sin08b] A. Singhee, Jiajing Wang, B.H. Calhoun, and R.A. Rutenbar, "Recursive Statistical Blockade: An Enhanced Technique for Rare Event Simulation with Application to SRAM Circuit Design", In *Proceedings International Conference on VLSI Design (VLSID)*, pages 131–136, 2008.

[Ste97] M. Steyaert, V. Peluso, J. Bastos, P. Kinget, and W. Sansen, "Custom analog low power design: the problem of low voltage and mismatch", In *Proceedings of IEEE Custom Integrated Circuits Conference (CICC)*, pages 285–292, 1997.

[Suz02] T. Suzuki, S. Nakahara, S. Iwahashi, K. Higeta, K. Kanetani, H. Nambu, M. Yoshida, and K. Yamaguchi, "Programmable and automatically-adjustable sense-amplifier activation scheme and multi-reset address-driven decoding scheme for high-speed reusable SRAM core", In *Proceedings IEEE Symposium on VLSI Circuits*, pages 44–45, 2002.

[Suz08] T. Suzuki, H. Yamauchi, Y. Yamagami, K. Satomi, and H. Akamatsu, "A Stable 2-Port SRAM Cell Design Against Simultaneously Read/Write-Disturbed Accesses", *IEEE Journal of Solid-State Circuits (JSSC)*, 43(9):2109–2119, 2008.

[Tak04] K. Takeda, Y. Hagihara, Y. Aimoto, M. Nomura, R. Uchida, Y. Nakazawa, Y. Hirota, S. Yoshida, and T. Saito, "Per-bit sense amplifier scheme for 1GHz SRAM macro in sub-100nm CMOS technology", In *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC)*, pages 502–542 Vol.1, 2004.

[Tak05] T. Takayanagi, J.L. Shin, B. Petrick, J.Y. Su, H. Levy, Ha Pham, J. Son, N. Moon, D. Bistry, U. Nair, and et al., "A dual-core 64-bit ultraSPARC microprocessor for dense server applications", *IEEE Journal of Solid-State Circuits (JSSC)*, 40(1):7–18, 2005.

[Tak06] K. Takeda, Y. Hagihara, Y. Aimoto, M. Nomura, Y. Nakazawa, T. Ishii, and H. Kobatake, "A read-static-noise-margin-free SRAM cell for low-VDD and high-speed applications", *IEEE Journal of Solid-State Circuits (JSSC)*, 41(1):113–121, 2006.

[Tra96] H. Tran, "Demonstration of 5T SRAM and 6T dual-port RAM cell arrays", In *Proceedings IEEE Symposium on VLSI Circuits*, pages 68–69, 1996.

[Ver07] M. Verma and P. Marwedel, *Advanced Memory Optimization Techniques for Low-Power Embedded Processors*, Springer Netherlands, 2007.

[Ver08a] Staf Verhaegen, Stefan Cosemans, Mircea Dusa, Pol Marchal, Axel Nackaerts, Geert Vandenberghe, Wim Dehaene, Vivek K. Singh, and Michael L. Rieger, "Litho variations and their impact on the electrical yield of a 32nm node 6T SRAM cell", In *Design for Manufacturability through Design-Process Integration II*, volume 6925, pages 69250R–12, San Jose, CA, USA, 2008. SPIE.

[Ver08b] N. Verma and A.P. Chandrakasan, "A 256 kb 65 nm 8T Subthreshold SRAM Employing Sense-Amplifier Redundancy", *IEEE Journal of Solid-State Circuits (JSSC)*, 43(1):141–149, 2008.

[Wan82] Niantsu Wang, "On the design of MOS dynamic sense amplifiers", *IEEE Transactions on Circuits and Systems*, 29(7):467–477, 1982.

[Wan04] A. Wang and A. Chandrakasan, "A 180mV FFT processor using subthreshold circuit techniques", In *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC)*, pages 292–529 Vol.1, 2004.

[Wan05a] H. Wang, M. Miranda, A. Papanikolaou, F. Catthoor, and W. Dehaene, "Variable tapered pareto buffer design and implementation allowing run-time configuration for low-power embedded SRAMs", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(10):1127–1135, 2005.

[Wan05b] C. Wann, R. Wong, D.J. Frank, R. Mann, Shang-Bin Ko, P. Croce, D. Lea, D. Hoyniak, Yoo-Mi Lee, J. Toomey, and et al., "SRAM cell design for stability methodology", In *Digest of Technical Papers IEEE International Symposium on VLSI Technology*, pages 21–22, 2005.

[Wan07a] Jiajing Wang and B.H. Calhoun, "Canary Replica Feedback for Near-DRV Standby VDD Scaling in a 90nm SRAM", In *Proceedings of IEEE Custom Integrated Circuits Conference (CICC)*, pages 29–32, 2007.

[Wan07b] Jiajing Wang, A. Singhee, R.A. Rutenbar, and B.H. Calhoun, "Statistical modeling for the minimum standby supply voltage of a full SRAM array", In *Proceedings European Solid-State Circuits Conference (ESSCIRC)*, pages 400–403, 2007.

[Wan08] Yih Wang, Hong Jo Ahn, U. Bhattacharya, Zhanping Chen, T. Coan, F. Hamzaoglu, W.M. Hafez, Chia-Hong Jan, P. Kolar, S.H. Kulkarni, and et al., "A 1.1 GHz 12 μA/Mb-Leakage SRAM Design in 65 nm Ultra-Low-Power CMOS Technology With Integrated Leakage Reduction for Mobile Applications", *IEEE Journal of Solid-State Circuits (JSSC)*, 43(1):172–179, 2008.

[Wan09] Y. Wang and e.a., "A 4.0 GHz 291Mb Voltage-Scalable SRAM Design in 32nm High-K Metal-Gate CMOS with Integrated Power Management", In *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC)*, pages 456–457, 2009.

[Wic04] B. Wicht, T. Nirschl, and D. Schmitt-Landsiedel, "Yield and speed optimization of a latch-type voltage sense amplifier", *IEEE Journal of Solid-State Circuits (JSSC)*, 39(7):1148–1158, 2004.

[Yam08] Masanao Yamaoka, Kenichi Osada, and Takayuki Kawahara, *A cell-activation-time controlled SRAM for low-voltage operation in DVFS SoCs using dynamic stability analysis*, pages 286–289, 2008.

[Yan05] Byung-Do Yang and Lee-Sup Kim, "A low-power SRAM using hierarchical bit line and local sense amplifiers", *IEEE Journal of Solid-State Circuits (JSSC)*, 40(6):1366–1376, 2005.

[Yos88] T. Yoshihara, H. Hidaka, Y. Matsuda, and K. Fujishima, "A Twisted Bit Line Technique for Multi-Mb Drams", In *Digest of Technical Papers IEEE International Solid-State Circuits Conference (ISSCC)*, pages 238–239, 1988.

[Zha05] K. Zhang, U. Bhattacharya, Zhanping Chen, F. Hamzaoglu, D. Murray, N. Vallepalli, Yih Wang, B. Zheng, and M. Bohr, "SRAM design on 65-nm CMOS technology with dynamic sleep transistor for leakage reduction", *IEEE Journal of Solid-State Circuits (JSSC)*, 40(4):895–901, 2005.

[Zha06] Kevin Zhang, U. Bhattacharya, Zhanping Chen, F. Hamzaoglu, D. Murray, N. Vallepalli, Yih Wang, Bo Zheng, and M. Bohr, "A 3-GHz 70-mb SRAM in 65-nm CMOS technology with integrated column-based dynamic power supply", *IEEE Journal of Solid-State Circuits (JSSC)*, 41(1):146–151, 2006.

[Zha08] Bo Zhai, S. Hanson, D. Blaauw, and D. Sylvester, "A Variation-Tolerant Sub-200 mV 6-T Subthreshold SRAM", *IEEE Journal of Solid-State Circuits (JSSC)*, 43(10):2338–2348, 2008.