

---

**Masterarbeit**

---

**von Frau Alena Larissa Weber**

**Entwurf eines Pixelsensorchips für die  
Teilchenphysik**

**Design of a Pixel Sensor Chip for Particle  
Physics**

Beginn: 19.04.2016  
Abgabe: 17.10.2016  
Betreuer: Prof. Dr. Ivan Peric  
Institut für Prozessdatenverarbeitung und  
Elektronik, IPE



Ich versichere hiermit, dass ich meine Masterarbeit selbständig und unter Beachtung der Regeln zur Sicherung guter wissenschaftlicher Praxis im Karlsruher Institut für Technologie (KIT) angefertigt habe.

Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und wörtlich oder inhaltlich übernommene Stellen als solche kenntlich gemacht.

Karlsruhe, den 17.10.2016

Alena Larissa Weber



## Zusammenfassung

Im Folgenden werden der Entwurf und die Optimierung eines Pixelsensorchips für die Teilchenphysik gezeigt. Zu Beginn werden die für das weitere Verständnis notwendigen Grundlagen erläutert und anschließend wird auf das Design des Pixelsensorchips eingegangen.

Es wird intensiv der Entwurf des analogen Teils gezeigt und die wichtigen neuen Ideen für Verbesserungen und neue Funktionen, sowie deren Realisierung, wie die beiden Modi des ADC, erläutert. Für den ADC wurde ein Konzept entworfen und realisiert, das im einen Modus zwei verschiedene konstante Schwellspannungen für zwei Komparatoren zur Verbesserung der Timewalk Problematik liefert und das im zweiten Modus eine Amplitudenmessung durch eine linear ansteigende Schwellspannung ermöglicht, die nur sehr wenig durch Rauschen beeinflusst wird.

Anschließend wird ausführlich das Design der Scanlogik gezeigt; hier wurde mithilfe von Gruppierungen und einem für jede Gruppe konstanten Signal aus der vorherigen Gruppe eine deutlich verkürzte Laufzeit erzielt.

Auch der digitale Schaltungsteil in der Peripherie des Pixelsensorchips wird gezeigt und die dort verwendete State Machine erklärt.

Einen großen Teil dieser Arbeit nimmt die Realisierung des Layouts des analogen und digitalen Teils des Pixelsensorchips ein. Hier werden Prinzipien, Ideen zur Optimierung und schließlich die erstellten Layouts gezeigt und erklärt.

# Inhaltsverzeichnis

<b>Zusammenfassung</b> .....	<b>i</b>
<b>Inhaltsverzeichnis</b> .....	<b>ii</b>
<b>Abbildungsverzeichnis</b> .....	<b>iv</b>
<b>Tabellenverzeichnis</b> .....	<b>vi</b>
<b>Abkürzungsverzeichnis</b> .....	<b>vii</b>
<b>1 Einführung</b> .....	<b>1</b>
1.1 Umfeld.....	1
1.2 Aufgabenstellung .....	2
<b>2 Grundlagen</b> .....	<b>3</b>
2.1 Transistor.....	3
2.1.1 Aufbau.....	3
2.1.2 Funktionsweise.....	4
2.2 Grundsaltungen.....	8
2.2.1 Verstärker mit Kaskode.....	8
2.2.2 Sourcefollower .....	9
2.3 HVCMOS Sensoren .....	10
2.4 Logische Elemente .....	11
2.4.1 Boolesche Algebra .....	11
2.4.2 and, or, not, nand, nor, xor .....	12
2.4.3 Realisierungsmöglichkeiten .....	15
2.5 Layout.....	17
2.5.1 Aufteilung in Elemente .....	17
2.5.2 Anordnen, Verbinden, Prüfen .....	17
<b>3 Analog Design</b> .....	<b>19</b>
3.1 Workflow .....	20
3.2 Element: Pixelverstärker .....	21
3.2.1 Konzept .....	21
3.2.2 Schaltplan .....	21
3.2.3 Simulation .....	22
3.3 Element: Signalübertragung zur Peripherie (Crosstalk).....	25
3.3.1 Konzept .....	25
3.3.2 Schaltplan .....	25
3.3.3 Simulation .....	31
3.4 Element: Analog Digital Converter (ADC) .....	35
3.4.1 Konzept .....	35
3.4.2 Schaltplan .....	35
3.4.3 Simulation .....	39
3.5 Zwei Modi für den ADC: zwei Schwellen-Modus und Amplitudenmessung-Modus	
42	
3.5.1 Konzept .....	42
3.5.2 Schaltplan .....	43

3.5.3	Simulation .....	44
3.6	Element: Scan/Oder-Kette mit fastIn .....	53
3.6.1	Konzept .....	53
3.6.2	Schaltplan .....	57
3.6.3	Simulation .....	58
<b>4</b>	<b>Digital Design.....</b>	<b>61</b>
4.1	Workflow .....	61
4.2	Aufbau des Digitalteils.....	63
4.2.1	Überblick.....	63
4.2.2	State Machine.....	63
<b>5</b>	<b>Layout.....</b>	<b>66</b>
5.1	Anforderungen .....	66
5.2	Ansatz für Optimierung.....	67
5.2.1	Idee .....	67
5.2.2	Umsetzung.....	67
5.2.3	Beispiele .....	68
5.3	Element: Komparator .....	70
5.3.1	Konzept .....	70
5.3.2	Umsetzung.....	70
5.4	Element: Logikteil nach dem ADC.....	71
5.4.1	Konzept .....	71
5.4.2	Umsetzung.....	71
5.5	Element: Scanlogik .....	72
5.5.1	Konzept .....	72
5.5.2	Umsetzung.....	72
5.6	Gesamtschaltung .....	73
<b>6</b>	<b>Anwendung.....</b>	<b>76</b>
6.1	Mu3e.....	76
6.1.1	Hintergründe und Ziel .....	76
6.1.2	Aufbau.....	76
<b>7</b>	<b>Zusammenfassung und Ausblick.....</b>	<b>78</b>
7.1	Zusammenfassung.....	78
7.2	Ausblick .....	78
<b>Anhang: Literatur.....</b>		<b>79</b>

## Abbildungsverzeichnis

Abbildung 1 Aufbau eines NMOS Transistors .....	3
Abbildung 2 Potentialenergie NMOS Transistor, sperrender Zustand .....	4
Abbildung 3 Potentialenergie NMOS Transistor, leitender Zustand .....	5
Abbildung 4 Transistor Kennlinie $I_{ds}$ über $V_{ds}$ .....	5
Abbildung 5 Schaltsymbole, links NMOS rechts PMOS .....	7
Abbildung 6 Kaskode .....	8
Abbildung 7 Spannungsverstärker mit Kaskode .....	9
Abbildung 8 Sourcefollower .....	9
Abbildung 9 Schematischer Aufbau eines HVCMOS Sensors .....	10
Abbildung 10 Symbol and .....	13
Abbildung 11 Symbol or .....	13
Abbildung 12 Symbol not .....	13
Abbildung 13 Symbol nand .....	14
Abbildung 14 Symbol nor .....	14
Abbildung 15 Symbol xor .....	15
Abbildung 16 CMOS not .....	16
Abbildung 17 CMOS nand .....	16
Abbildung 18 CMOS nor .....	17
Abbildung 19 Flowconcept Pixelsensorchip .....	19
Abbildung 20 Aufbau Pixelsensorchip .....	19
Abbildung 21 Analog Design Flow .....	20
Abbildung 22 Schaltplan Amplifier .....	21
Abbildung 23 Simulation Amplifier .....	22
Abbildung 24 Monte Carlo Simulation Amplifier .....	23
Abbildung 25 Rauschsimulation Amplifier .....	24
Abbildung 26 Schaltplan Amplifier Crosstalk .....	25
Abbildung 27 Amplifier mit Sendeschaltung .....	26
Abbildung 28 Empfangsschaltung .....	27
Abbildung 29 Schaltplan Amplifier Crosstalk mit Sende- und Empfangsteil .....	28
Abbildung 30 Amplifier mit Sendeteil von zwei komplementären Signalen .....	29
Abbildung 31 Schaltplan Amplifier Crosstalk Sende- und Empfangsteil, Variante mit zwei komplementären Signalen .....	30
Abbildung 32 Simulation Amplifier Crosstalk .....	31
Abbildung 33 Simulation Amplifier Crosstalk Variante mit Stromübertragung .....	32
Abbildung 34 Simulation Amplifier mit differentieller Stromübertragung Crosstalk .....	33
Abbildung 35 Simulation Amplifier mit differentieller Stromübertragung Crosstalk, groß ...	33
Abbildung 36 Vorgängerversion des Komparators .....	36
Abbildung 37 Optimierung des Komparators, Teil 1 .....	37
Abbildung 38 Optimierung des Komparators, Teil 2 .....	38
Abbildung 39 Komparator mit einer Schwelle .....	39

---

Abbildung 40 Komparator mit einer Schwelle, vergrößert.....	39
Abbildung 41 Monte Carlo Simulation Komparator.....	40
Abbildung 42 Rauschsimulation Komparator.....	41
Abbildung 43 Schaltplan, zwei Modi für ADC.....	44
Abbildung 44 Simulation Komparator zwei Schwellen.....	44
Abbildung 45 Simulation zwei Schwellen bei verschiedenen Eingangsspannungen, Verbesserung des Timewalk Problems .....	45
Abbildung 46 Simulation Komparator mit linear steigender Schwellspannung.....	46
Abbildung 47 Simulation Komparator mit linear steigender Schwellspannung, Sweep mit drei verschiedenen Amplituden.....	47
Abbildung 48 Monte Carlo Simulation Modus mit zwei Schwellspannungen.....	48
Abbildung 49 Monte Carlo Simulation Modus mit linear steigender Schwellspannung.....	49
Abbildung 50 Rauschsimulation des Modus I mit zwei konstanten Schwellspannungen.....	50
Abbildung 51 Rauschsimulation des Modus II mit Amplitudenmessung.....	51
Abbildung 52 Rausch-Simulation im Modus II: Amplitudenmessung mit verschiedenen Eingangssignalamplituden .....	52
Abbildung 53 OR-Kaskadierung.....	53
Abbildung 54 NOR/NAND-Kaskadierung .....	55
Abbildung 55 Scanlogik mit vier Eingängen .....	55
Abbildung 56 Scanlogik, zwei Gruppen mit je acht Eingängen .....	57
Abbildung 57 Simulation Scanlogik, ohne fastIn, drei Gruppen mit je acht Elementen .....	58
Abbildung 58 Simulation Scanlogik, mit fastIn, drei Gruppen mit je 8 Eingängen .....	59
Abbildung 59 Monte Carlo Simulation der Scanlogik.....	60
Abbildung 60 Workflow digital Design.....	61
Abbildung 61 Ablauf von Place & Route .....	62
Abbildung 62 Überblick Digitalteil der Chipperipherie .....	63
Abbildung 63 State Machine.....	65
Abbildung 64 Anordnung der Elemente für den digital Teil zwischen Komparator und Scanlogik.....	68
Abbildung 65 Anordnung der Elemente für die Komparatoren mit beiden Modi .....	68
Abbildung 66 Anordnung der Elemente für die Scanlogik.....	69
Abbildung 67 Layout Komparator .....	70
Abbildung 68 Layout digital Teil.....	71
Abbildung 69 Layout Scanlogik .....	72
Abbildung 70 Links: Layout der Elemente ohne Scanlogik, rechts: Layout mit Scanlogik ....	74
Abbildung 71 Layout einer Gruppe (20 Pixelelemente) .....	75
Abbildung 72 Schematischer Aufbau des Mu3e Experiments [3].....	76
Abbildung 73 Vergrößerte Darstellung des Aufbaus des Mu3e Experiments [3] .....	77

## Tabellenverzeichnis

Tabelle 1 Wahrheitstabelle and .....	12
Tabelle 2 Wahrheitstabelle or .....	13
Tabelle 3 Wahrheitstabelle not.....	13
Tabelle 4 Wahrheitstabelle nand .....	14
Tabelle 5 Wahrheitstabelle nor .....	14
Tabelle 6 Wahrheitstabelle xor .....	15
Tabelle 7 Beispiel für Funktion der Scanlogik.....	53
Tabelle 8 Beispiel für Ablauf der State Machine .....	65

## Abkürzungsverzeichnis

$\mu$	Leitwert
A	Verstärkung
ADC	Analog-Digital-Wandler
CMOS	Complementary Metal-Oxide-Semiconductor
$C_{ox}$	Kapazität des Oxids
DRC	Design Rules Check
eoc	end of column = Ende der Spalte
gm	Gain
HVCMOS	High Voltage CMOS
$I_{DS}$	Strom von Drain nach Source
$I_{DS, Sat}$	Strom von Drain nach Source bei Sättigung
$I_{in}$	Strom an Eingang der Schaltung
$I_{out}$	Strom am Ausgang der Schaltung
L	Länge
LVS	Layout versus Schematic
NMOS	n-Type Metal-Oxide Semiconductor
PMOS	p-Type Metal-Oxide Semiconductor
$r_{DS}$	Widerstand zwischen Drain und Source
$R_{in}$	Eingangswiderstand
VDD	Versorgungsspannung
$V_{DS}$	Spannung zwischen Drain und Source
$V_{GS}$	Spannung zwischen Gate und Source
$V_{th}$	Threshold Voltage = Schwellspannung
W	Weite



# 1 Einführung

## 1.1 Umfeld

Die Halbleiterindustrie wächst seit Jahren stetig und es können immer mehr Transistoren auf immer dünneren Chips realisiert werden. Gerade in der Halbleiterfertigung werden große Fortschritte gemacht, die dazu führen, dass die Möglichkeiten des realisierbaren Chipdesigns beständig wachsen. Die Entwicklung der Halbleitertechnologie kann mit Moores Law beschrieben werden: seit den 1970er Jahren verdoppelt sich alle 18 Monate die maximale Anzahl von Komponenten (Transistoren) auf einem IC (Integrated Circuit).

Pixelsensoren finden heute ein breites Anwendungsgebiet, u.a. in Konsumartikeln wie Handykameras und Digitalkameras, in medizinischen Bildgebungsverfahren, in Sensoren für die Teilchenphysik oder für Materialwissenschaften. Die Pixelsensoren für die wissenschaftliche Anwendung müssen andere Anforderungen erfüllen als die in den Konsumgütern eingesetzten. In der wissenschaftlichen Anwendung werden niedriges Rauschen, Einzel-Partikel-Detektion, 100% Fill Factor, eine hohe Zeitauflösung und hohe Strahlungstoleranz gefordert.

Eine neue Familie der Pixelsensoren bilden die High-Voltage Complementary Metal-Oxide Semiconductor (HVCMOS). Der im Folgenden vorgestellte Pixelchip gehört dieser Familie an. Im Unterschied zu den Standard-Pixelsensoren, die die Elektronik nahe den Detektorelementen realisiert haben, ist die Pixelelektronik bei HVCMOS in der Sensorelektrode realisiert. Dies führt u.a. zu dem Vorteil, dass 100% Fill Factor erreicht wird.

Bei dem Mu3e Chip handelt es sich um einen Halbleiter-Pixelsensor. Die Vorteile der Halbleitersensoren im Vergleich zu älteren elektronischen Sensoren sind die schnelle Antwortzeit, die niedrige Schwellenergie, niedrigere Fertigungskosten und natürlich die Möglichkeit, sehr viele kleine Sensoreinheiten auf einem einzigen Chip zu realisieren. [4]

## 1.2 Aufgabenstellung

Ziel dieser Arbeit ist der Entwurf eines Pixelsensorchips für die Teilchenphysik. Die Anforderungen richten sich dabei nach dem Mu3e Projekt des Paul Scherrer Instituts in der Schweiz. Der hier entworfene Chips soll in erster Linie für dieses Experiment verwendet werden.

Die Größe des Chips soll 2cm x 2cm betragen und die Größe eines Pixels  $80\mu\text{m} \times 80\mu\text{m}$ . Wichtig sind eine sehr gute örtliche Auflösung und eine gute zeitliche Auflösung von unter 100ns eines Treffers.

Ein Schwerpunkt liegt hier bei der Verbesserung des Timewalk-Problems. Es sollen eine oder mehrere Lösungen zur deutlichen Verbesserung dieses Problems entworfen und realisiert werden.

Es soll eine Möglichkeit gefunden werden, zuverlässig die Amplitude des Signals, welches vom Pixel zum analogen und digitalen Teil des Readout-Elements jedes Pixels geführt wird, zu bestimmen.

Die Scanlogik soll neu entworfen werden und dabei die Signallaufzeit durch die Scanlogik deutlich verkürzt werden.

Es sollen verschiedene Simulationsreihen durchgeführt und ausgewertet werden, beispielsweise zum Mismatch der Transistoren und zum Rauschen.

Im Anschluss an den Schaltungsentwurf, die Simulationen und Optimierungen soll das Layout in der ams18 Technologie entworfen werden unter Beachtung aller Anforderungen und Design Regeln.

## 2 Grundlagen

### 2.1 Transistor

Zunächst eine kurze Beschreibung des zentralen Bauteils eines Chips: des Transistors. Die Metal-Oxide-Semiconductor-Field-Effect Transistoren (MOSFET) sind die meist verwendeten Transistoren, sie werden zur Realisierung von CMOS (Complementary Metal-Oxide-Semiconductor) Schaltungen verwendet. Eine CMOS Schaltung besteht immer aus einem Teil mit NMOS Transistoren (n-type Metal-Oxide Semiconductor) und einem komplementären Teil mit PMOS (p-type Metal-Oxide Semiconductor) Transistoren. Die CMOS Transistoren sind die Erfindung, die die Computer-Revolution ermöglicht hat und deren Skalierbarkeit sie in der bekannten hohen Geschwindigkeit vorantreibt.

Transistoren können zwei verschiedene Funktionen erfüllen: zum einen als elektrischer Schalter, der die Umsetzung von logischen Funktionen ermöglicht, oder zum anderen als Verstärker in analogen Schaltungen.

#### 2.1.1 Aufbau

CMOS Transistoren werden auf Silicium-Wafer realisiert. Durch verschieden dotierte Bereiche und Siliciumoxidschichten wird ein Transistor gebildet. Auf diesen Aufbau soll nun eingegangen werden.

Der Aufbau wird an einem NMOS Transistor erläutert. Nach der Erklärung des NMOS Transistors werden unten die Unterschiede zu einem PMOS Transistor vorgestellt.

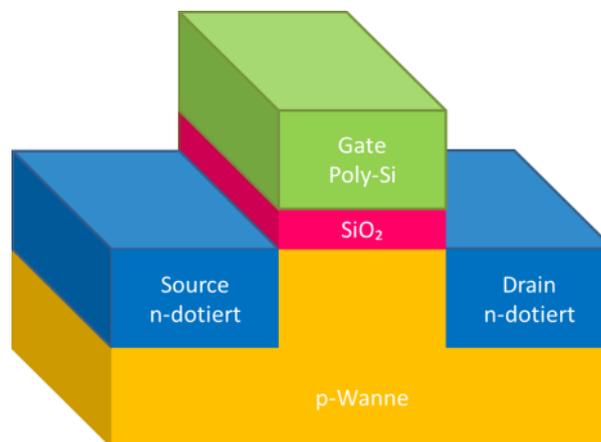


Abbildung 1 Aufbau eines NMOS Transistors

Abbildung 1 zeigt den Querschnitt eines NMOS Transistors. Dieser ist umgeben von p-dotiertem Silicium, der p-Wanne. Über dieser p-Wanne befinden sich zwei Bereiche aus n-dotiertem Silicium, diese werden als Source und Drain bezeichnet. Über der p-Wanne zwischen Source und Drain befindet sich eine dünne Schicht Siliciumoxid. Diese dünne Isolatorschicht ist entscheidend für die korrekte Funktion des Transistors, daher sollte diese von sehr guter Qualität ohne Verunreinigungen sein. Darüber befindet sich das sogenannte Gate aus Poly-Silicium. An Source, Gate und Drain befinden sich Metallkontakte. Umgeben wird die gesamte Struktur von Feldoxid (Siliciumdioxid) zur elektrischen Isolation.

Der Unterschied zu einem PMOS Transistor besteht nur darin, dass hier Source und Drain p – dotiert sind und sich der ganze Transistor in einer n – Wanne befindet.

### 2.1.2 Funktionsweise

Zunächst die Aufgaben von Source, Gate und Drain. Die Source stellt die Quelle für freie Ladungsträger dar, bei einem NMOS Transistor sind dies die Elektronen, bei einem PMOS Transistor die Löcher. Drain sammelt die Ladungsträger und das Gate dient zur Ansteuerung. Der Transistor kann entweder als Verstärker oder als Schalter, dessen Stromfluss durch eine Spannung kontrolliert wird, eingesetzt werden.

Leitet der MOSFET Transistor einen Strom, so ist dies immer ein Oberflächenstrom, da er nur unmittelbar unter dem Isolator fließt.

Die Funktionsweise wird zuerst an einem NMOS Transistor erklärt, anschließend werden die Unterschiede zu einem PMOS Transistor erklärt.

Im NMOS Transistor sind Elektronen die Ladungsträger. Liegt an einem MOSFET keine Spannung an, so befindet er sich immer im sperrenden Zustand. Da Source und Drain dotiert sind, haben sie ein höheres Potential als das Substrat. Das höhere Potential von Source und Drain wird durch die Elektronen erzeugt, während sich im Substrat zwischen Source und Drain vor allem Löcher befinden. Durch diesen Potentialunterschied entsteht eine sogenannte Potentialbarriere. Abbildung 2 zeigt schematisch den Verlauf der Potentialenergie. Es wird deutlich, dass keine Ladungsträger diese Energie aufbringen können, um die Barriere überwinden zu können, es kann also kein Strom fließen.

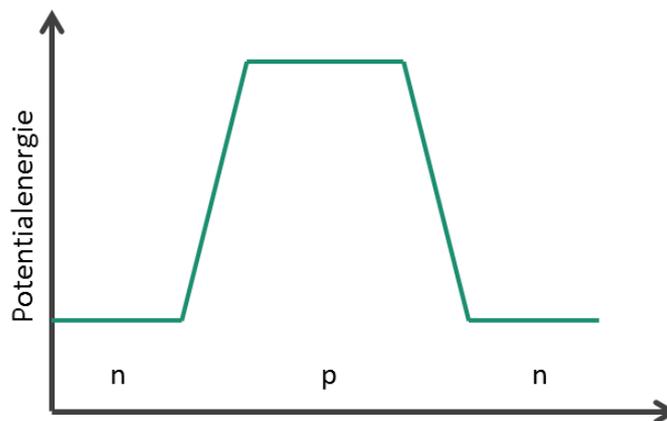


Abbildung 2 Potentialenergie NMOS Transistor, sperrender Zustand

Legt man nun jedoch eine Spannung zwischen Gate und Source an,  $U_{GS}$ , so kommt es zu einem elektrischen Feld in dem Substrat zwischen Drain und Source. Durch dieses Feld werden die Löcher aus dieser Substratzone noch unten in das tiefere Substrat verdrängt, es entsteht eine Verarmungszone. Nun ist die Potentialenergie, die überwunden werden muss, damit es zu einem Stromfluss kommt, deutlich kleiner. Die Elektronen können, falls die Spannung ausreichend groß ist und damit das Feld ausreicht, um die Potentialbarriere ausreichend zu verkleinern, durch einen Elektronenkanal an der Silicium-Oxid-Oberfläche von Source nach Drain fließen, siehe Abbildung 3. Die notwendige Spannung  $U_{GS}$ , die man

anlegen muss, um so einen Stromfluss zwischen Source und Drain zu ermöglichen, wird als Threshold Voltage oder Schwellspannung bezeichnet. Liegt nun also eine ausreichend große Spannung an, so kommt es zu einem Oberflächenstrom zwischen Source und Drain, direkt unterhalb der Isolatorschicht des Gates. Genauer handelt es sich um einen Driftstrom.

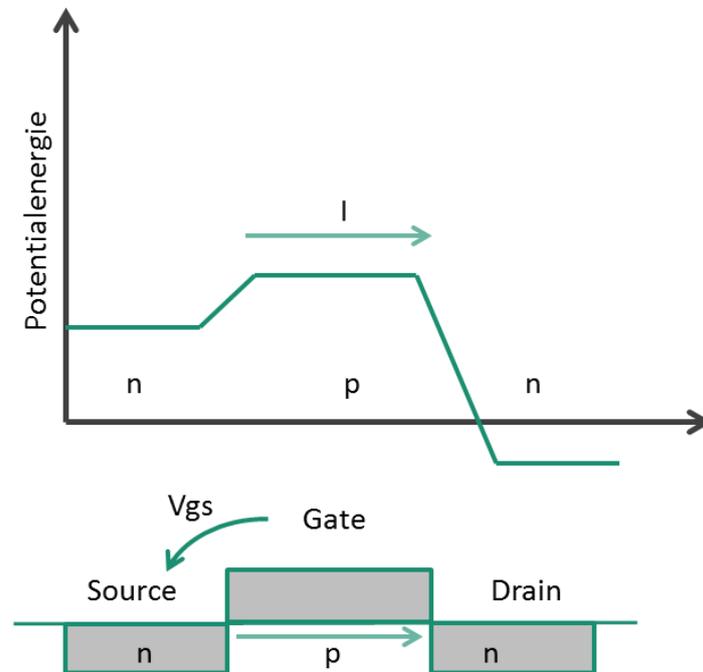


Abbildung 3 Potentialenergie NMOS Transistor, leitender Zustand

Nachdem nun die Funktionsweise erklärt wurde, werden die Transistorkennlinien mit den dazugehörigen Formeln erläutert. Es werden hier die üblichen genäherten Formeln verwendet, für eine genaue Betrachtung ohne diese Näherungen sei auf [2] verwiesen.

Zu Beginn die Kennlinie des Ausgangstroms  $I_{ds}$ , zu sehen in Abbildung 4.

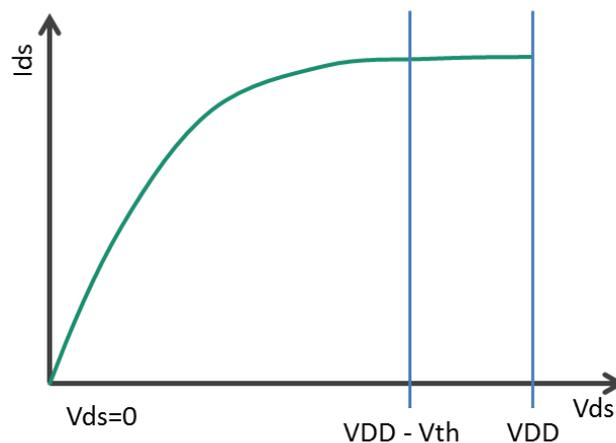


Abbildung 4 Transistor Kennlinie  $I_{ds}$  über  $V_{ds}$

Hier erkennt man, dass sich der Transistor wie ein spannungsabhängiger Widerstand verhält. Unten die Formel für die  $I_{DS}$  Kennlinie. Anschließend werden nun die einzelnen Teile der Kennlinie und die Faktoren genauer erläutert.

$$I_{DS} = \mu \cdot C_{ox} \cdot \frac{W}{L} \cdot \left( (V_{GS} - V_{th}) \cdot V_{DS} - \frac{V_{DS}^2}{2} \right)$$

Ist  $V_{GS} = 0$  so ist auch  $I_{DS} = 0$  für alle  $V_{DS}$ , wie oben gesehen muss  $V_{GS} > V_{TH}$  sein, um einen  $I_{DS}$  Strom zu ermöglichen, hier ist jedoch  $V_{GS} < V_{TH}$  und somit natürlich kein Stromfluss möglich.

Der vordere lineare Teil der Kennlinie wird beschrieben durch folgende Formel:

$$I_{DS} = \mu \cdot C_{ox} \cdot \frac{W}{L} \cdot (V_{GS} - V_{th}) \cdot V_{DS}$$

Der hintere lineare Teil der Kennlinie, also im Bereich  $V_{GS} > V_{DS} - V_{TH}$  verläuft nach folgender Formel:

$$I_{DS, sat} = \frac{1}{2} \cdot \mu \cdot C_{ox} \cdot \frac{W}{L} \cdot (V_{GS} - V_{th})^2$$

Hier befindet sich der Transistor in Sättigung. Für einen bestimmten Stromwert ist der Widerstand am Kanalende unendlich und der Strom kann daher nicht mehr weiter steigen.

Der Faktor  $V_{GS} - V_{th}$  beschreibt, dass es bei einer höheren  $V_{GS}$  mehr Elektronen im Kanal gibt, folglich steigt der Strom in diesem Fall.

Die Faktoren  $\frac{W}{L}$  und  $\mu$  beschreiben den Leitwert und dessen Abhängigkeit von Mobilität und Geometrie. Bei einer größeren Stromfläche ( $W \cdot T$ ), wobei  $T$  die Dicke repräsentiert, kann mehr Strom fließen, gleichzeitig führt ein längerer Kanal, also ein größerer Wert für  $L$ , zu weniger Strom.

Der Faktor  $\frac{V_{DS}^2}{2}$  kommt aus der Überlegung, dass ein Anstieg von  $V_{DS}$  eine Erhöhung des Widerstands bewirkt und damit der Strom langsamer steigt.

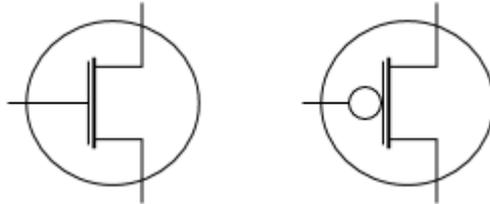
Auch gibt es noch den Faktor  $C_{ox}$  in der oben gezeigten Formel, dies ist die Oxid-Kapazität pro Fläche, ein hohes  $C_{ox}$  bewirkt mehr Ladung im Kanal.

Hier gilt es noch anzumerken, dass die Mobilität der Löcher zweimal niedriger ist als die der Elektronen, daher leitet ein PMOS Transistor bei gleichem  $\frac{W}{L}$  Verhältnis immer 2x schlechter als ein NMOS Transistor.

Nun die weiteren Unterschiede zwischen PMOS und NMOS. Bei einem NMOS Transistor sind Source und Drain n-dotiert, bei einem PMOS Transistor dagegen sind Source und Drain p-dotiert. Damit ein NMOS Transistor leitet, ist eine positive Gate-Source-Spannung notwendig. Diese bewirkt eine Verdrängung der Löcher, es kommt zu einem Elektronenkanal. Die Voraussetzung für diesen leitenden Zustand ist  $V_{GS} > V_{TH}$ . An Source muss daher ein niedrigeres Potential anliegen. Im Gegensatz dazu werden bei einem PMOS Transistor

Elektronen durch eine negative Gate-Source Spannung verdrängt, es kommt zur Bildung eines Löcherkanals, der den Stromfluss ermöglicht, also gilt  $V_{GS} < V_{TH}$ . An der Source befindet sich ein höheres Potential.

Abbildung 5 zeigt die beiden Schaltsymbole für einen NMOS- und für einen PMOS Transistor.



**Abbildung 5** Schaltsymbole, links NMOS rechts PMOS

## 2.2 Grundsaltungen

Im Folgenden werden drei einfache Grundsaltungen vorgestellt, die später für das analog Design verwendet wurden.

### 2.2.1 Verstärker mit Kaskode

Zuerst werden kurz die Eigenschaften der Kaskode erläutert und anschließend wird auf eine Verstärkerschaltung mit Kaskode eingegangen.

Abbildung 6 zeigt eine Kaskode. Die Kaskode funktioniert als ein Impedanzwandler, wobei der Ausgangsstrom  $I_{out}$  ungefähr dem Eingangsstrom  $I_{in}$  entspricht. Außerdem ist der Eingangswiderstand durch  $R_{in} = \frac{1}{gm_{casc}}$  gegeben und damit klein. Der Ausgangswiderstand ist dagegen groß  $R_{out} > 10 \cdot rds_{sig}$  und wird beschrieben durch

$$R_{out} = gm_{casc} \cdot rd_{casc} \cdot rds_{sig}$$

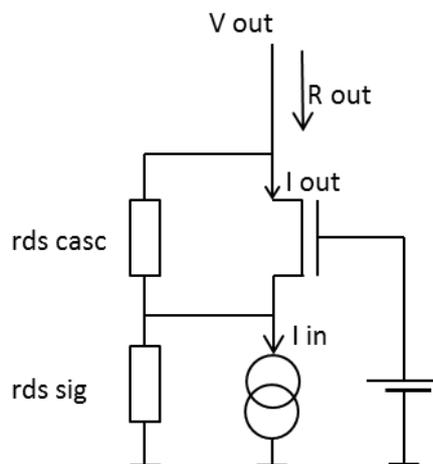


Abbildung 6 Kaskode

Eine einfache Verstärkerschaltung mit Kaskode ist in Abbildung 7 gezeigt. Die Verstärkung wird durch die gleiche Formel gegeben wie bei der Schaltung ohne Kaskode:

$$A = -I_{ds} \cdot R_{out} = -gm \cdot V_{gs} \cdot R_{out}$$

Nun ist jedoch  $R_{out}$  in der Schaltung mit Kaskode höher als in der ohne. Ohne Kaskode ist der Ausgangswiderstand  $rds$ , mit Kaskode beträgt der Ausgangswiderstand:

$$R_{out} = gm_{casc} \cdot rds_{casc} \cdot rds$$

Der Ausgangswiderstand der Schaltung mit Kaskode ist mindestens um den Faktor 10 größer als bei der Schaltung ohne Kaskode. Damit ist die Spannungsverstärkung mit Kaskode gegeben durch

$$A = -gm \cdot (rds_{load} \parallel gm_{casc} \cdot rds_{casc} \cdot rds)$$

Ohne Kaskode ist die Spannungsverstärkung

$$A = -gm (rds_{load} \parallel rds)$$

Typischerweise ist daher die Verstärkung mit Kaskode um mindestens den Faktor 10 größer, bei gleichem  $gm$ . [5]

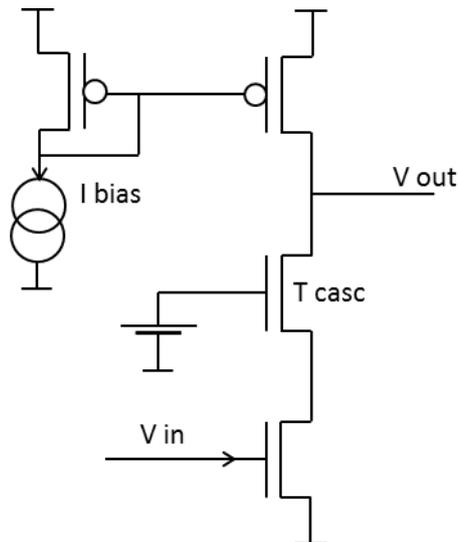


Abbildung 7 Spannungsverstärker mit Kaskode

### 2.2.2 Sourcefollower

Der Sourcefollower oder auch Common Drain Verstärker ist in Abbildung 8 zu sehen. Der Eingang ist an Gate und der Ausgang an Source des Transistors angeschlossen. An Source ist zusätzlich eine Stromquelle als Biaselement angeschlossen. Dies kann durch einen NMOS Transistor oder durch einen hohen Widerstand realisiert werden. Die gebräuchlichere Variante ist dabei der NMOS Transistor. Beim Sourcefollower dient der Transistor als Verstärker. Der Transistor ist, wie in 2.1 erklärt, ähnlich einer spannungsgesteuerten Stromquelle mit einem Innenwiderstand von  $r_{ds}$ .

Der DC Wert am Ausgang des Sourcefollowers ist um  $V_{th} - V_{ds, sat}$  kleiner als der DC Wert am Ausgang der Schaltung. Die AC Verstärkung beträgt ungefähr eins, was bedeutet, dass das Ausgangssignal auch bei Anwesenheit eines AC Signals um einen konstanten Wert kleiner ist als das Potential am Eingang. Nun wird auch klar, warum diese Schaltung als Sourcefollower bezeichnet wird: die Spannung am Ausgang folgt auf gewisse Weise der Eingangsspannung.

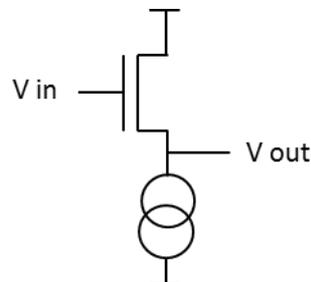


Abbildung 8 Sourcefollower

## 2.3 HVCMOS Sensoren

Der Sensor des vorgestellten Chips für die Detektion eines Teilchen ist in der HVCMOS (High Voltage CMOS) Technologie entworfen. Diese Technologie wird seit 2006 entwickelt und seit 2010 angewendet, unter anderem für das Mu3e Experiment.

Das Besondere im Vergleich zu herkömmlichen CMOS Sensoren ist, dass der Sensor und die Elektronik nicht räumlich getrennt sind, sondern „ineinander“ liegen. Im Substrat gibt es eine tiefe n-Wanne, siehe Abbildung 9. Diese n-Wanne bildet gemeinsam mit dem Substrat eine n-p- Struktur wie bei einer Diode. Dies n-Wanne / Substrat Diode dient als Sensorelement. In dieser n-Wanne liegen sowohl eine p-Wanne als auch eine weitere n-Wanne für die Sensorelektronik in CMOS Technologie. Ein HVCMOS Sensor erlaubt damit eine komplexe Elektronikschaltung für den Sensor, die direkt im Sensor realisiert werden kann. Man spricht von high Voltage, da eine hohe Spannung verwendet wird, um das Substrat um die tiefe n-Wanne zu verarmen.

Ein großer Vorteil der HVCMOS Sensortechnik ist, dass die Sensorchips in einer für die Industrie gebräuchlichen Technik hergestellt werden können: der HVCMOS-Technik. Daher sind Pixelsensorchips in HVCMOS günstig in der Herstellung und bedürfen keiner extra Anpassung des Fertigungsprozesses. [4]

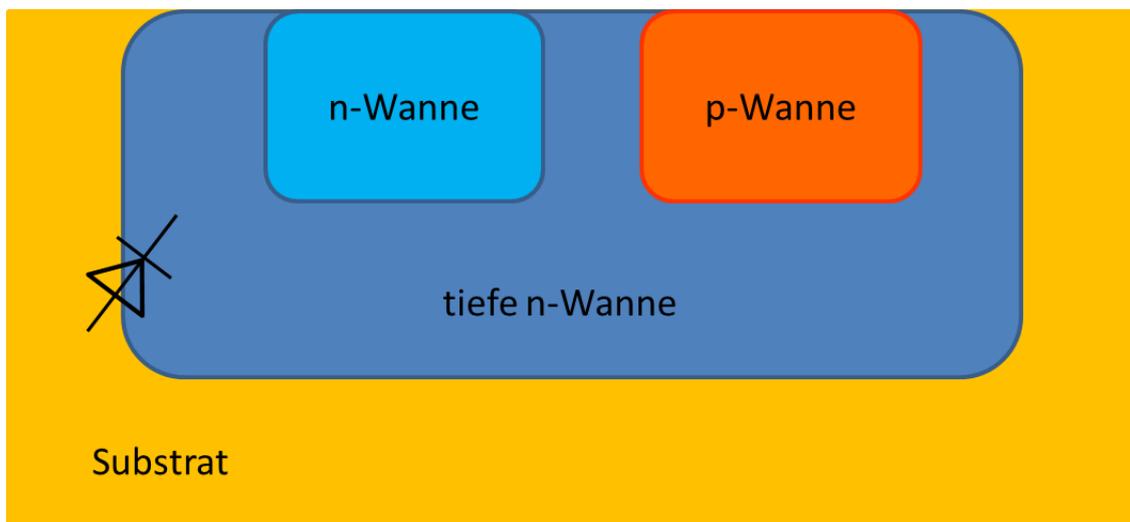


Abbildung 9 Schematischer Aufbau eines HVCMOS Sensors

## 2.4 Logische Elemente

### 2.4.1 Boolesche Algebra

Für die boolesche Algebra gelten die Huntington'schen Axiome. Im Folgenden werden diese kurz vorgestellt, für einen tieferen Einblick sei auf [1] verwiesen. Daran anschließend werden Möglichkeiten zur Vereinfachung vorgestellt.

Zunächst werden eine Menge  $K$

und zwei Operatoren  $\blacktriangle$  und  $\blacktriangledown$  definiert.

$a, b, c, I, O, A \in K$ .

H1: „Abgeschlossenheit“

Werden die Operatoren auf Elemente der Menge  $K$  angewendet, so führt diese Anwendung nicht aus der Menge  $K$  heraus.

$$a \blacktriangle b \in K$$

$$a \blacktriangledown b \in K$$

H2: „Kommutativ-Gesetz“

Die Reihenfolge, in der die Operanden angeordnet sind, ist vertauschbar, ohne dass sich das Ergebnis ändert.

$$a \blacktriangle b = b \blacktriangle a$$

H3: „Distributiv-Gesetz“

Die beiden Operatoren  $\blacktriangle$  und  $\blacktriangledown$  distribuieren über den anderen.

$$(a \blacktriangle b) \blacktriangledown c = (a \blacktriangledown c) \blacktriangle (b \blacktriangledown c)$$

$$(a \blacktriangledown b) \blacktriangle c = (a \blacktriangle c) \blacktriangledown (b \blacktriangle c)$$

H4: „Existenz eines neutralen Elements“

Zu jedem Operator gibt es ein sogenanntes neutrales Element. Wendet man den Operator in Kombination mit dem zugehörigen neutralen Element auf ein beliebiges Element der Menge  $K$  an, so bleibt dieses unverändert.

$$a \blacktriangle O = a$$

$$a \blacktriangledown I = a$$

H5: „Komplement“

Es existiert zu jedem Element  $a$  aus  $K$  ein komplementäres Element ebenfalls aus  $K$ . Wird das komplementäre Element mit  $a$  verknüpft, so erhält man das neutrale Element.

$$a \blacktriangle A = O$$

$$a \blacktriangledown A = I$$

## Schaltalgebra

Die Schaltalgebra ist eine boolesche Algebra. Sie besteht aus einer binären Wertemenge mit den Elementen 0 und 1. Da die Wertemenge binär ist, sind die neutralen Elemente gleich diesen beiden Werten der Menge. Die Schaltalgebra enthält folgende Operatoren: and, or, not, xor und die zusammengesetzten Operatoren nand, nor, xnor.

Zusätzlich zu den Huntington'schen Axiomen gelten für die Schaltalgebra noch weitere Regeln, die jedoch auf die Huntington'schen Axiome zurückgehen oder von diesen abgeleitet werden. Von diesen Regeln werden hier lediglich zwei vorgestellt, da diese in 3.6 verwendet wurden.

„Absorptionsgesetz“

$$a \vee (a \wedge b) = a$$

$$a \wedge (a \vee b) = a$$

Dies lässt sich einfach überprüfen: auf der linken Seite: ist der gegebene Ausdruck von a oder von der Verknüpfung a und b abhängig, so lässt sich dies vereinfachen zu der Aussage, dass lediglich a überprüft werden muss, um das gewünschte Ergebnis zu erhalten. Die rechte Seite lässt sich ähnlich einfach prüfen, es muss auf jeden Fall a = 1 sein, um den linken Teil des & zu erfüllen, dann ist selbstverständlich auch das V in der Klammer erfüllt. Daher lässt sich auch dieser Ausdruck zu a vereinfachen.

„De Morgansche Regel“

$$\overline{(a \vee b)} = \bar{a} \wedge \bar{b}$$

$$\overline{(a \wedge b)} = \bar{a} \vee \bar{b}$$

Werden zwei Variablen mit einem and (or) verknüpft und der gesamte Ausdruck negiert, so lässt sich dies immer umformen zu der Negation der einzelnen Variablen verknüpft durch ein or (and).

### 2.4.2 and, or, not, nand, nor, xor

Im Folgenden werden die grundlegenden logischen Operationen and, or, not, nor, nand und xor beschrieben.

Zunächst das and: das and beschreibt eine Konjunktion. Diese erreicht den Wert 1 nur, wenn alle Variablen, die durch das and verknüpft wurden, den Wert 1 haben. Folglich ist das and 0 dominant. Es gibt verschiedene Schreibweisen für den and Operator: „&“, „•“ oder „^“. Die Wahrheitstabelle für and mit zwei Variablen zeigt Tabelle 1. Unter der Wahrheitstabelle befindet sich im Folgenden immer das zu der logischen Operation gehörende Symbol.

<b>a</b>	<b>b</b>	<b>a &amp; b</b>
0	0	0
0	1	0
1	0	0
1	1	1

**Tabelle 1 Wahrheitstabelle and**

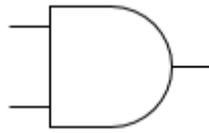


Abbildung 10 Symbol and

Die nächste grundlegende logische Funktion ist das or. Das or wird auch als Disjunktion bezeichnet. Werden Variablen mit einem or verknüpft, so erhalten diese den Wert 0 nur in dem Fall, dass alle Variablen 0 sind. Man spricht von einer 1 Dominanz. Die verschiedenen Schreibweisen für ein or sind „|“, „+“ oder „V“. Tabelle 2 zeigt die Wahrheitstabelle für ein or mit zwei Variablen.

$a$	$b$	$a \vee b$
0	0	0
0	1	1
1	0	1
1	1	1

Tabelle 2 Wahrheitstabelle or

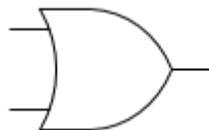


Abbildung 11 Symbol or

Das not repräsentiert eine Negierung. Eine 1 mit einem not ergibt somit eine 0 und eine 0 kombiniert mit not eine 1. Das not ist von sehr großer Bedeutung für die boolesche Logik und für die Umformungsregeln. Auch für das not gibt es unterschiedliche Schreibweisen, diese werden nun je in Verbindung mit der Variablen  $a$  gezeigt: „! $a$ “ oder „ $\bar{a}$ “. Die folgende Tabelle 3 zeigt die Wahrheitstabelle des not.

$a$	$\bar{a}$
0	1
1	0

Tabelle 3 Wahrheitstabelle not

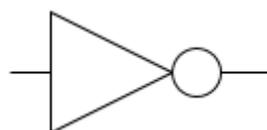


Abbildung 12 Symbol not

Das nand ist die negierte Form des and. Das bedeutet, die Ergebnisse der and Verknüpfung werden negiert und so eine neue Funktion erzeugt: das nand. Das nand ergibt daher nur 0,

falls alle Variablen den Wert 1 hatten. Die Schreibweise für das nand setzt sich aus den Zeichen für and und not zusammen: „ $\bar{\&}$ “, „! $\&$ “ oder „ $\bar{V}$ “. Die Tabelle 4 zeigt zur Veranschaulichung sowohl das nand als auch das and für zwei Variable in Form einer Wahrheitstabelle.

$a$	$b$	$a \& b$	$a \bar{\&} b$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Tabelle 4 Wahrheitstabelle nand

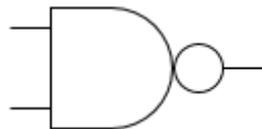


Abbildung 13 Symbol nand

Nach dem gleichen Prinzip wie das nand wird auch das nor gebildet. Das nor erzeugt das gleiche Ergebnis wie das or, nur in negierter Form. Das bedeutet, es wird nur dann eine 1 erzeugt, wenn alle Variablen den Wert 0 haben. Analog zu dem nand wird auch das nor Symbol aus den Symbolen für or und not gebildet. Auch hier in Tabelle 5 die Wahrheitstabelle von nor und or.

$a$	$b$	$a \vee b$	$a \bar{\vee} b$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Tabelle 5 Wahrheitstabelle nor

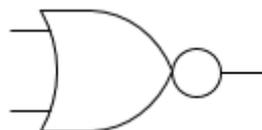


Abbildung 14 Symbol nor

Die letzte wichtige Funktion ist das xor, auch bezeichnet als Antivalenz. Das xor stellt ein exklusives or dar. Also ein or, das nur dann erfüllt wird, sprich eine 1 zeigt, falls es sich zwar um ein or handelt, aber nicht gleichzeitig auch ein and erfüllt wird. Für zwei Variable muss folglich stets eine Variable 0 sein und die andere 1, um eine wahre Aussage zu erhalten.

<i>a</i>	<i>b</i>	<i>a xor b</i>
0	0	0
0	1	1
1	0	1
1	1	0

Tabelle 6 Wahrheitstabelle xor

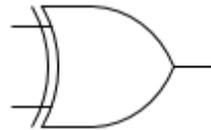


Abbildung 15 Symbol xor

### 2.4.3 Realisierungsmöglichkeiten

In dieser Arbeit wird die CMOS Technologie verwendet. Diese ist heute eine der am meist verbreiteten Technologien. Die CMOS Schaltungen haben mehrere Vorteile, zum einen erlauben sie eine sehr hohe Integrationsdichte und zum anderen verhindern sie einen kontinuierlichen Stromfluss, hiermit sind in erster Linie Querströme im durchgeschalteten Zustand der Transistoren gemeint. Zusätzlich sind CMOS Schaltungen sehr schnell, da die Umladezeit nur vom Widerstand des leitenden Transistors abhängt. [1] [9]

Zur Realisierung von CMOS Schaltungen werden die oben genannten logischen Funktionen benötigt. Es lassen sich mit wenigen Elementen sogenannte Basissysteme bilden. Mithilfe der Elemente eines Basissystems lassen sich alle Operatoren der Schaltalgebra bilden. Es bestehen verschiedene Möglichkeiten, aus den Operatoren diejenigen für ein Basissystem zu bilden.

In dieser Arbeit wurde das am häufigsten verwendete Basissystem gewählt, in diesem sind die Grundoperatoren nand, nor und not enthalten. Mithilfe dieser drei Elemente lassen sich alle Funktionen der Schaltalgebra abbilden. Dies soll im Folgenden kurz gezeigt werden:

and: Kombination von nand und not

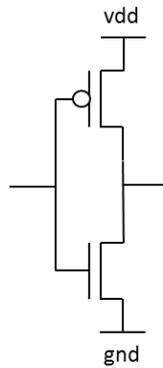
or: Kombination von nor und not

exnor: das exnor wird repräsentiert durch folgenden Ausdruck  $(a \& b) \vee (\bar{a} \& \bar{b})$ , hier können nun die and und or Elemente durch die zuvor beschriebenen Kombinationen ersetzt werden.

exor: hier handelt es sich um ein negiertes exnor, es lässt sich daher ebenfalls mit den Grundfunktionen nor, nand und nor darstellen.

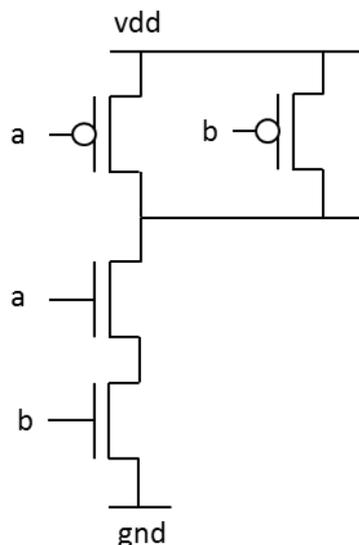
Nun reicht es folglich, aus den Grundelementen CMOS Schaltungen zu realisieren, diese können anschließend zu jedem gewünschten Ausdruck kombiniert werden.

Zunächst die einfachste CMOS Schaltung der logischen Elemente: das not. Das not besteht aus einem NMOS und einen PMOS Transistor, wie in Abbildung 16 gezeigt. Liegt am Eingang eine 0, also low, an, so leitet der obere Transistor, der PMOS Transistor, und der Ausgang wird daher ebenfalls auf high gezogen. Liegt dagegen am Eingang ein high an, so leitet der NMOS Transistor und der PMOS Transistor befindet sich in sperrenden Zustand. Dadurch liegt am Ausgang Masse, also 0, an. Somit wurde mithilfe dieser einfachen Schaltung ein not realisiert.



**Abbildung 16 CMOS not**

Als nächstes die nand CMOS Schaltung, diese ist in Abbildung 17 zu sehen. Es werden je zwei PMOS- und NMOS-Transistoren verwendet. Auch hier lässt sich die Korrektheit der Schaltung durch ähnliche Überlegungen wie bei not überprüfen. Man betrachtet für jede Zeile der Wahrheitstabelle die Schaltung, je nach den Eingängen leiten verschiedene Transistoren, was zu dem gewünschten Ausgangssignal führt.



**Abbildung 17 CMOS nand**

Analog dazu lässt sich auch das nor in CMOS realisieren. Die Schaltung hierfür ist in Abbildung 18 gezeigt. Die Überlegungen zum Aufbau der Schaltung sind analog zu den bereits beschriebenen und werden daher nun nicht nochmals erläutert.

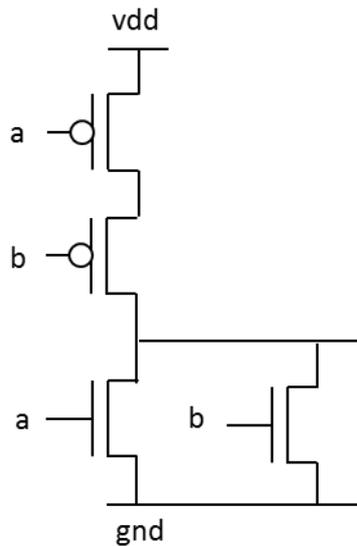


Abbildung 18 CMOS nor

## 2.5 Layout

### 2.5.1 Aufteilung in Elemente

Das Design des Layouts ist ein sehr umfangreicher Schritt des Chipdesigns. Es müssen verschiedene Vorgaben eingehalten werden und eine Realisierung auf kleinstmöglichem Raum durchgeführt werden. Hierfür ist ein systematisches Vorgehen von Vorteil. Das wichtigste Prinzip, welches hier berücksichtigt werden muss, ist „divide and conquer“, „teile und herrsche“. Dieses Prinzip wird ebenfalls beim Design des Schaltplans angewendet. Es bedeutet, dass man den Pixelsensorchip in viele kleine Elemente aufteilt, die man später wieder zu einer Einheit zusammensetzt. Hier wird zunächst das kleinste sich wiederholende Element ermittelt und anschließend wird dieses in funktionale Blöcke erneut geteilt. Diese sind hier beispielsweise der Pixelsensor, der ADC, die Scanlogik etc. Zusätzlich wird hier auch eine weitere Teilung vorgenommen: die logischen Gatter werden alleine für sich im Layout realisiert und anschließend als feste Blöcke, die nur in Ausnahmefällen erneut verändert werden, für zum Beispiel die Scanlogik, verwendet.

### 2.5.2 Anordnen, Verbinden, Prüfen

Für jedes Element wird als eigene Einheit ein Layout erstellt und anschließend geprüft. Folgende Regeln haben sich während dieser Masterarbeit als grundlegend für den Layoutprozess gezeigt. Zunächst sollten, je weiter „unten“ im gesamten Layout ein Element steht, auch nur die unteren Metalllagen verwendet werden. Werden die Grundlogikelemente wie Inverter oder And oder dergleichen designt, so sollte, wann immer möglich, Polysilicium zur Verbindung verwendet werden. Signale sind zwar mit Polysilicium deutlich langsamer als bei Metallverbindungen, aber auf dieser Ebene sind die Verbindungen sehr, sehr kurz. Daher ist es hier möglich Polysilicium zu verwenden. Auf der nächsten Ebene sollte ausschließlich

die erste Metalllage benutzt werden, nur falls dies nicht möglich ist, sollte die zweite Metalllage Verwendung finden.

Dies ist ein wichtiger Grundsatz: ist eine Verbindung mit einer tieferen Metalllage möglich, so sollte immer diese verwendet werden und nur falls dies absolut unmöglich ist, auf die nächst höhere zurückgegriffen werden. Auch sollte darauf geachtet werden, eine Metalllage primär für horizontale Verbindungen zu verwenden und die nächste Lage primär für vertikale Verbindungen. Dies bewirkt eine Struktur, die sich nicht selbst, bzw. den Designer durch konfuse Verbindungen in den möglichen Wegoptionen blockiert.

Als erstes wird ein Design Rules Check durchgeführt. Dieser prüft, ob das Design alle vorgegebenen Regeln bezüglich der Struktur, wie Abstände oder minimale Fläche, einhält. Die Design Rules sind für jede Technologie verschieden und richten sich daher direkt an die spätere Realisierung des Chips. Werden hier Fehler gefunden, muss das Layout überarbeitet werden. Nachdem der Design Rules Check, kurz DRC, erfolgreich war, kann die nächste Prüfung erfolgen: der Layout vs. Schematic Check, kurz LVS. Hier wird das Layout gegenüber dem Schematic getestet und geprüft, ob alle Elemente und Verbindungen vorhanden sind. Auch hier muss bei einer Fehlerdetektion das Layout überarbeitet werden. Nach jeder Überarbeitung muss zunächst wieder ein DRC erfolgen und dann ein LVS Check.

Nachdem diese beiden Tests durchgeführt wurden, ist das Layout für dieses Element abgeschlossen. Im nächsten Schritt erfolgt ein Zusammenfügen der Elemente, die in funktionelle Gruppen geteilt wurden, zu einem neuen, großen Layout. Um später einen LVS Check durchführen zu können, sollte auch ein Schematic für dieses neue, größere Element erstellt werden. Im Layout werden alle notwendigen Verbindungen nach den oben bereits aufgeführten Empfehlungen (möglichst niedriges Layer verwenden, kurze Verbindungen, beachten, ob dieses Layer für vertikale oder horizontale Verbindungen bevorzugt verwendet wurde) gemacht. Ist das Layout für dieses größere Element abgeschlossen, wird erneut ein DRC und LVS durchgeführt. Dieser Vorgang wird solange wiederholt, bis das Layout für das kleinste sich wiederholende Element fertig ist. Dieses kann nun periodisch wiederholt werden, um das gesamte Layout einer Spalte zu erzeugen. Hierbei kann es unter Umständen von Vorteil sein, das Element immer an einer Achse zu spiegeln. Bei dem Layout des hier gezeigten Pixelsensorchips wurde das Layout ohne die Scanlogik immer an der horizontalen Achse gespiegelt. Da sich immer zwei Elemente eine Scanlogik teilen (siehe 3.6.2), kann diese dann bei jedem Element an der gleichen Stelle platziert werden. Die Verbindungen, die benötigt werden, um die Elemente zu verbinden, können ebenfalls als ein extra Element realisiert werden, damit man auch dieses einfach periodisch wiederholen kann, um die gesamte Spalte zu verbinden. So lässt sich systematisch ein periodisches Layout auf optimierte Weise erzeugen.

### 3 Analog Design

Hier soll nun auch eine kurze Vorstellung des gesamten Pixelchips erfolgen, um einen ersten Überblick über die einzelnen Teile zu geben.

Der Pixelsensorchip besteht aus drei Teilen. Zuerst gibt es den Pixelsensor, dieser besteht aus dem HVCMOS Sensor und einem Verstärker. Als nächstes folgt der ADC (Analog Digital Converter) mit einem digitalen Logikteil. Der Pixelsensor und der ADC sind in mehreren Spalten angeordnet. Jede Spalte besteht aus 200 Elementen. Insgesamt ergibt sich so ein Pixelarray zur Detektion und Bestimmung des Ortes des Treffers. Am Ende jeder Spalte gibt es einen weiteren Digitalteil, der auch die end of colum, kurz eoc, enthält.

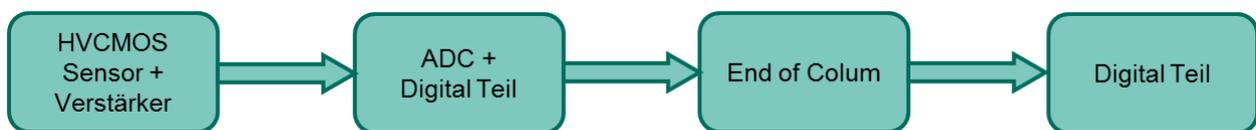


Abbildung 19 Flowconcept Pixelsensorchip

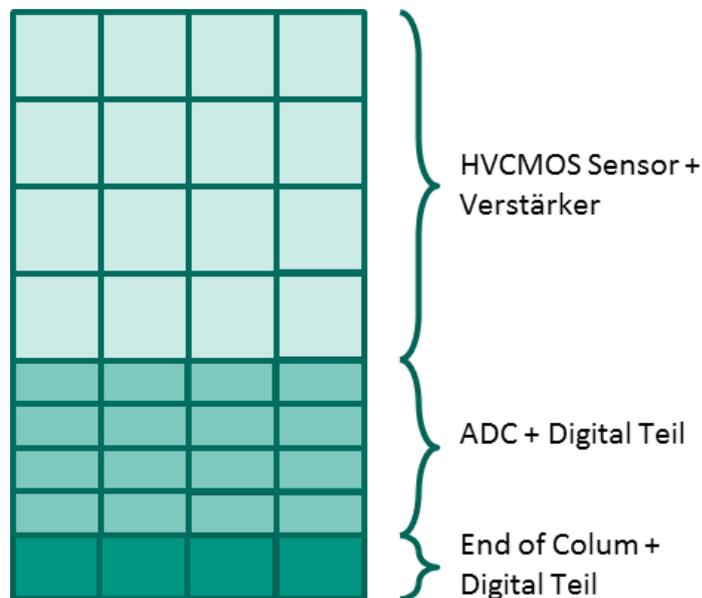


Abbildung 20 Aufbau Pixelsensorchip

### 3.1 Workflow

Im Folgenden wird der Ablauf des analogen Workflow vorgestellt, nach diesem sind unter anderem der ADC und die Scanlogik entwickelt worden. Bevor man jedoch mit diesem Workflow anfängt, sollte man das Prinzip „teile und herrsche“ berücksichtigen und den gesamten Chip in mehrere kleine Einheiten teilen. Diese werden dann zunächst alleine für sich entworfen und anschließend erfolgt eine Integration. Nun aber zu dem Workflow: Im ersten Schritt muss man sich über die Spezifikationen klar werden und diese festhalten. Anschließend erfolgt das Schematic Design, also der direkte Entwurf des Schaltplans. Danach wird die Schaltung simuliert. Diese Simulationsergebnisse werden ausgewertet und es wird geprüft, ob der Entwurf die zuvor festgelegten Spezifikationen erfüllt. Wenn dies nicht der Fall ist, muss eine Überarbeitung des Schematic Design erfolgen. Stimmt die Simulation mit den Anforderungen überein, so erfolgt im nächsten großen Schritt das Design des Layouts. Daran anschließend werden die Design Rules geprüft. Diese sind durch den Prozess, in welchem der Chip später gefertigt wird, festgelegt. Im Pixelsensorchip für das Mu3e Projekt ist es der AMS H18 Prozess. Kommt es zu einem Problem beim Prüfen der Design Rules, muss das Layout überarbeitet werden und der Design Rules Check erneut ausgeführt werden. Sind hier alle Fehler behoben, erfolgt der nächste Schritt: das Layout wird gegen die Schematic getestet, kurz LVS. Auch hier erfolgt eine Überarbeitung des Layouts, wenn es zu Fehlern beim Check kommt. Mit der Überarbeitung müssen selbstverständlich auch erneut die Design Rules geprüft werden. Sind der DRC und LVS erfolgreich, so erfolgt eine erneute Simulation basierend auf dem Layout. Erfüllt diese die Spezifikationen, so ist der Design Prozess abgeschlossen und es kommt zum Tape-out. Gibt es dagegen Unstimmigkeiten mit den Spezifikationen, erfolgt eine erneute Überarbeitung des Layouts.

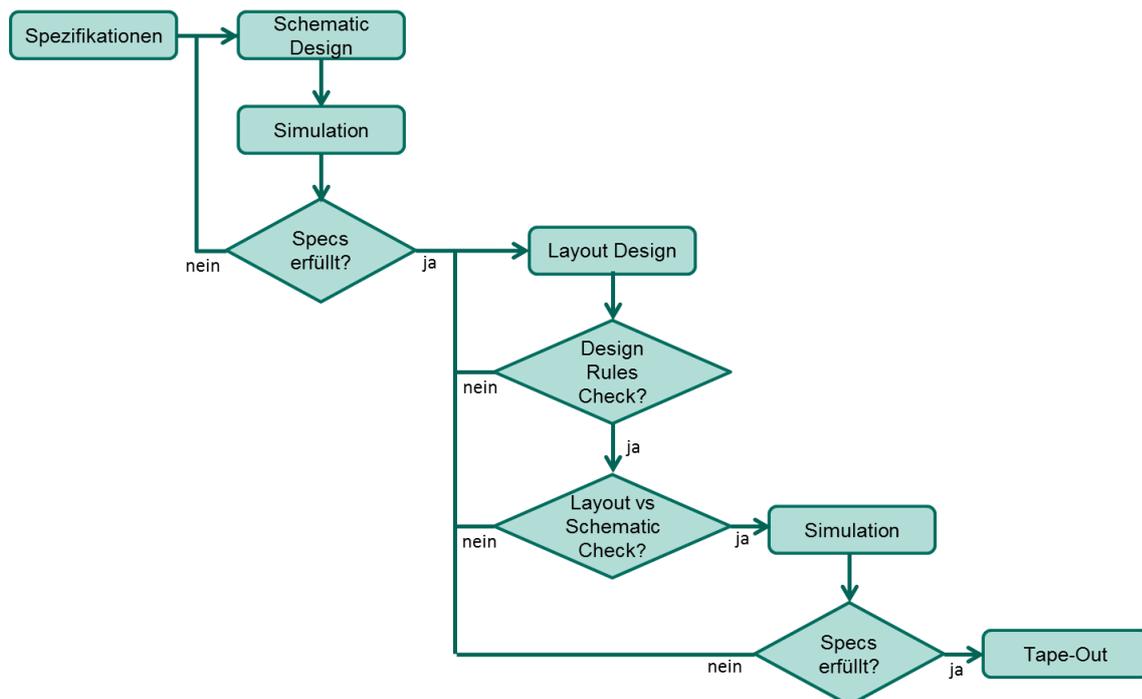


Abbildung 21 Analog Design Flow

## 3.2 Element: Pixelverstärker

### 3.2.1 Konzept

Der Pixelverstärker ist ein Verstärker (Amplifier), der direkt nach dem Pixelsensor und vor den beiden Komparatoren geschaltet wird. Die Aufgabe des Pixelverstärkers ist es, das noch recht schwache Signal aus dem Sensor zu verstärken. Hier möchte man die Amplitude deutlich vergrößern und gleichzeitig ist darauf zu achten, dass die Anstiegszeit minimal wird, dies ermöglicht eine schnelle Schaltung. Auch sollte die Abklingzeit klein sein, damit möglichst schnell das nächste Signal verstärkt werden kann.

Der Pixelverstärker soll mithilfe einer Kaskode realisiert werden, siehe 2.2.1. Zusätzlich soll die Schaltung mithilfe einer Rückkopplung verbessert werden. Für den Verstärker sind vor allem auch die  $\frac{W}{L}$  Verhältnisse der Transistoren, sprich die richtige Dimensionierung der Schaltung, sehr wichtig. Die erste Schätzung der Größen erfolgt durch eine ungefähre Berechnung, anschließend bietet sich eine Reihe von Veränderungen und Simulationen an, um systematisch durch kleine Änderungen der Transistoren das Optimum für diesen Anwendungsfall zu finden. Daher wurden speziell für dieses Element Simulationsreihen durchgeführt.

### 3.2.2 Schaltplan

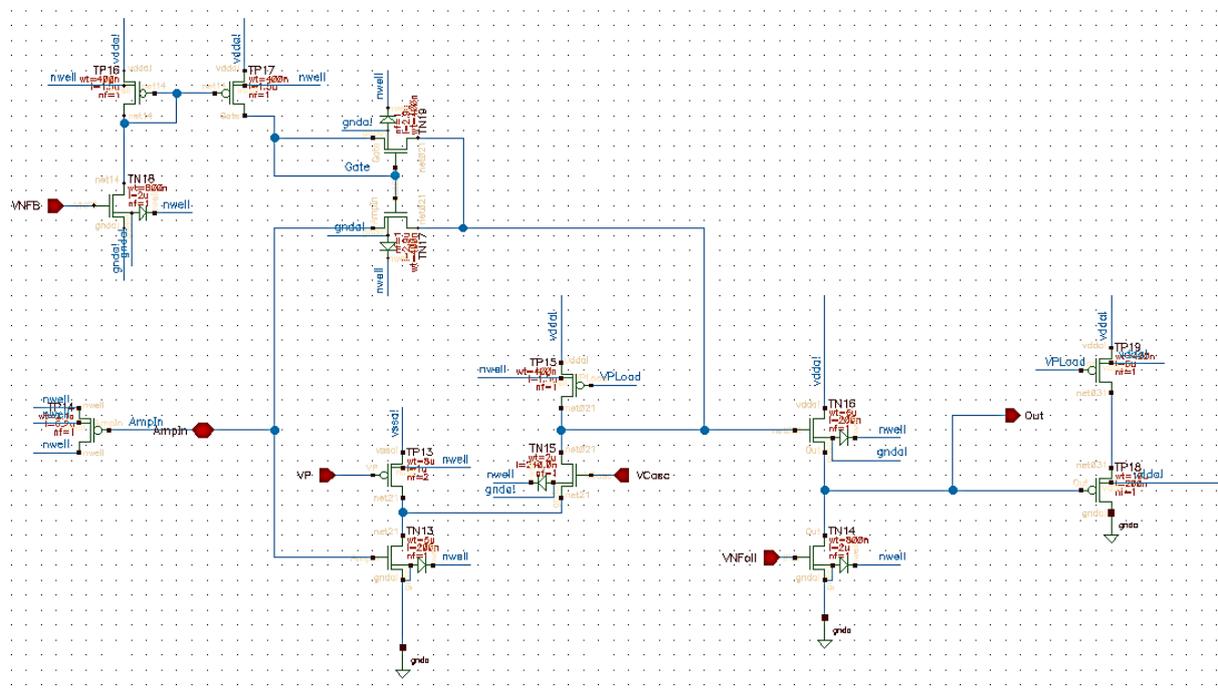


Abbildung 22 Schaltplan Amplifier

Der Schaltplan in Abbildung 22 zeigt den Amplifier. Im linken Teil der Schaltung befindet sich die in 2.2.1 vorgestellte Kaskode. Mithilfe der Kaskode lässt sich eine große Verstärkung

realisieren. Die Rückkopplung des Ausgangssignals zurück in den vorderen Bereich der Schaltung befindet sich im mittleren Teil der Schaltung. Der Ausgang (Out) liegt ganz rechts. In diesem Anwendungsfall hat sich gezeigt, dass die hier dargestellte Rückkopplung, also nicht ganz am Ende der Schaltung beginnend, ein besonders gutes Ergebnis bei der Simulation erzielt.

### 3.2.3 Simulation

#### 3.2.3.1 Ideale Simulation

Das Ergebnis der Simulation des Verstärkers ist in Abbildung 23 gezeigt. Die rote Kennlinie zeigt den Ausgang des Verstärkers. Die steigende Flanke zeigt einen steilen Anstieg und entspricht den Anforderungen. Die hellgrüne Kurve zeigt den Spannungsabfall in der n-Wanne, an die der Verstärker angeschlossen ist und die als Input dient.

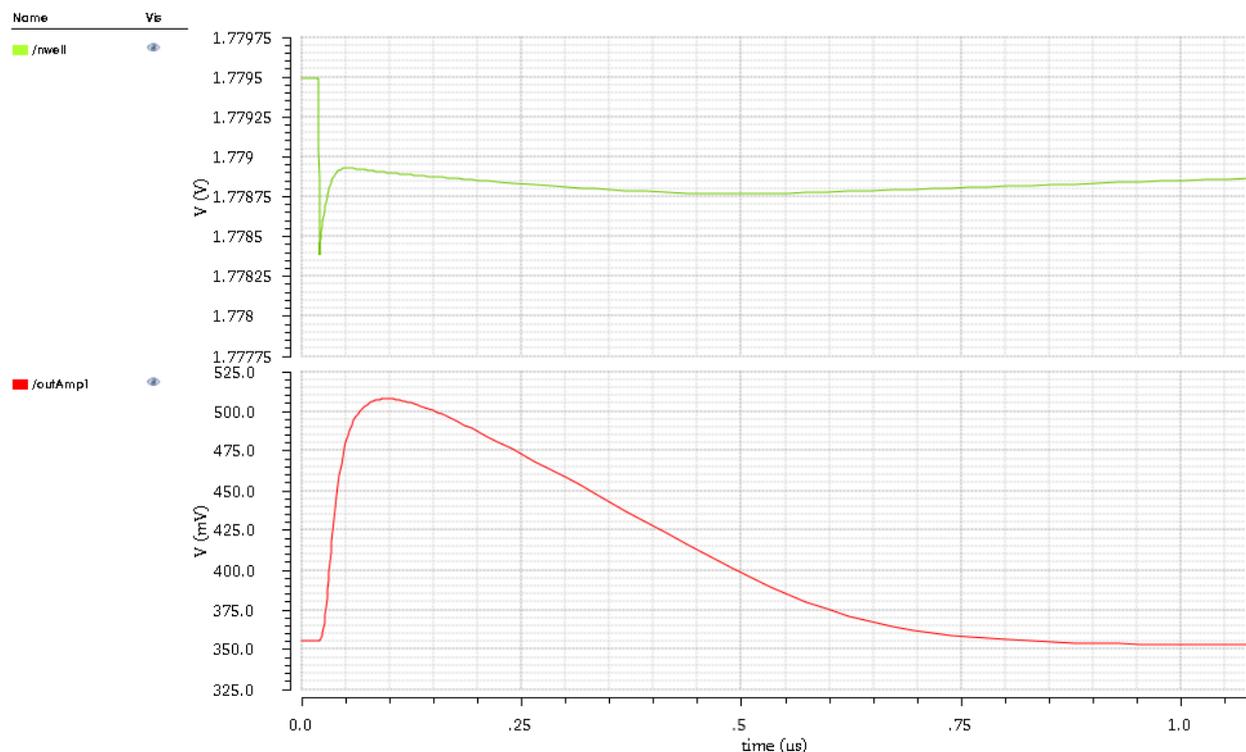


Abbildung 23 Simulation Amplifier

### 3.2.3.2 Monte Carlo Simulation

Die Monte Carlo Simulation zeigt den Einfluss von dem Mismatch der Transistoren auf ihr Verhalten. Durch den Fertigungsprozess entstehen immer Abweichungen der Transistoren, es handelt sich folglich nicht mehr um ideale Bauteile. Diese Abweichungen werden als Mismatch bezeichnet. Die Schaltung sollte eine Toleranz in Bezug auf diesen Mismatch zeigen. In Abbildung 24 ist das Ergebnis dieser Simulation gezeigt. Im kritischen Teil des Outputs, der steigenden Flanke, hat der Mismatch praktisch keine Auswirkung. Die Abweichungen bei der fallenden Flanke sind für die Schaltung ohne große Bedeutung. Insgesamt ist es daher ein sehr gutes Ergebnis.

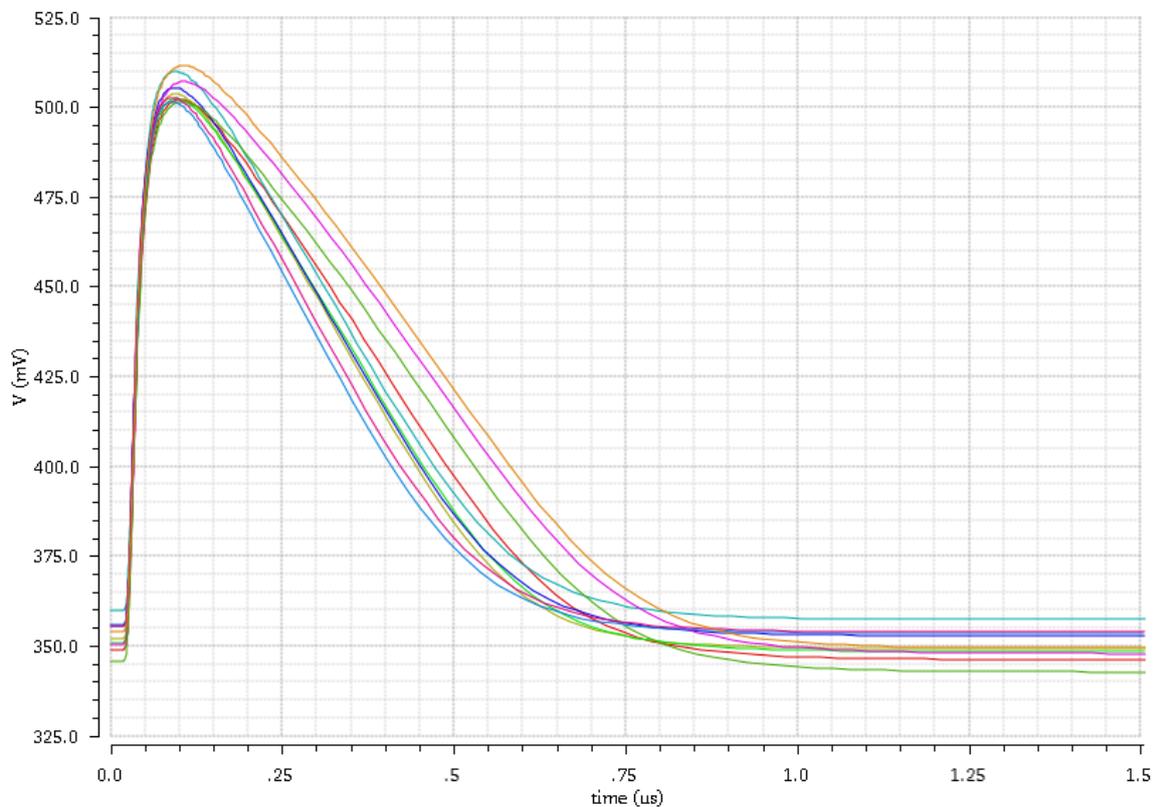


Abbildung 24 Monte Carlo Simulation Amplifier

### 3.2.3.3 Simulation mit Rauschen

Die nächste wichtige Simulation ist die mit Rauschen. Hier wird der Einfluss des Rauschens auf die Funktionalität der Schaltung simuliert. Hier gilt ähnliches, wie zuvor beim Monte Carlo Verfahren beschrieben, das Rauschen sollte den kritischen, bedeutenden Teil der Schaltung nicht zu stark negativ beeinflussen. Auch hier konnte mit dem Amplifier ein gutes Ergebnis erzielt werden, siehe Abbildung 25. Die steigende Flanke bleibt trotz des Rauschens praktisch unverändert im Vergleich zu den rauschfreien, idealen Simulationen in Abbildung 23.

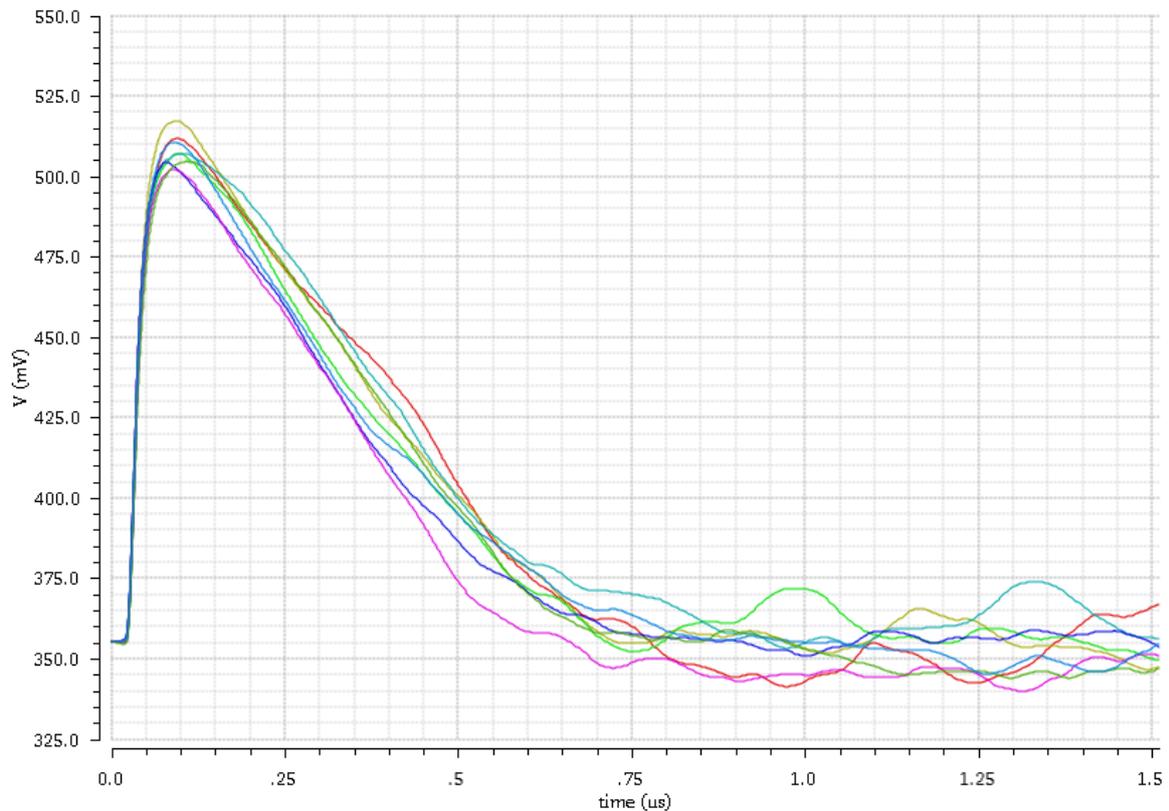


Abbildung 25 Rauschsimulation Amplifier

### 3.3 Element: Signalübertragung zur Peripherie (Crosstalk)

Im Zuge dieser Arbeit wurde das Crosstalk-Verhalten des Amplifier näher untersucht. Das Signal muss vom Ausgang des Pixelverstärkers zu dem analogen Pixelteil mit den Komparatoren und allen weiteren Elementen übertragen werden. Diese Übertragung erfolgt von jedem Pixelverstärker parallel zu den Übertragungskanälen der anderen Verstärker. Hierbei kann es passieren, dass ein Signal ein schwaches weiteres Signal auf der benachbarten Signalleitung verursacht; ein sogenannter Crosstalk: Diese Verhalten ist natürlich unerwünscht und sollte möglichst vermieden werden.

#### 3.3.1 Konzept

Es wurden verschiedene Variationen des Pixelverstärkers entworfen. Die Variationen unterscheiden sich unter anderem am Punkt der Rückkopplung oder beispielsweise darin, ob eine aktive Regelung zur Unterdrückung des Crosstalk Effekts beiträgt. Die verschiedenen Realisierungsmöglichkeiten wurden simuliert und mithilfe der Simulationsergebnisse weiter optimiert. Nach einer Analyse der Simulationsergebnisse wurde die für diese Arbeit beste Variante ausgewählt.

#### 3.3.2 Schaltplan

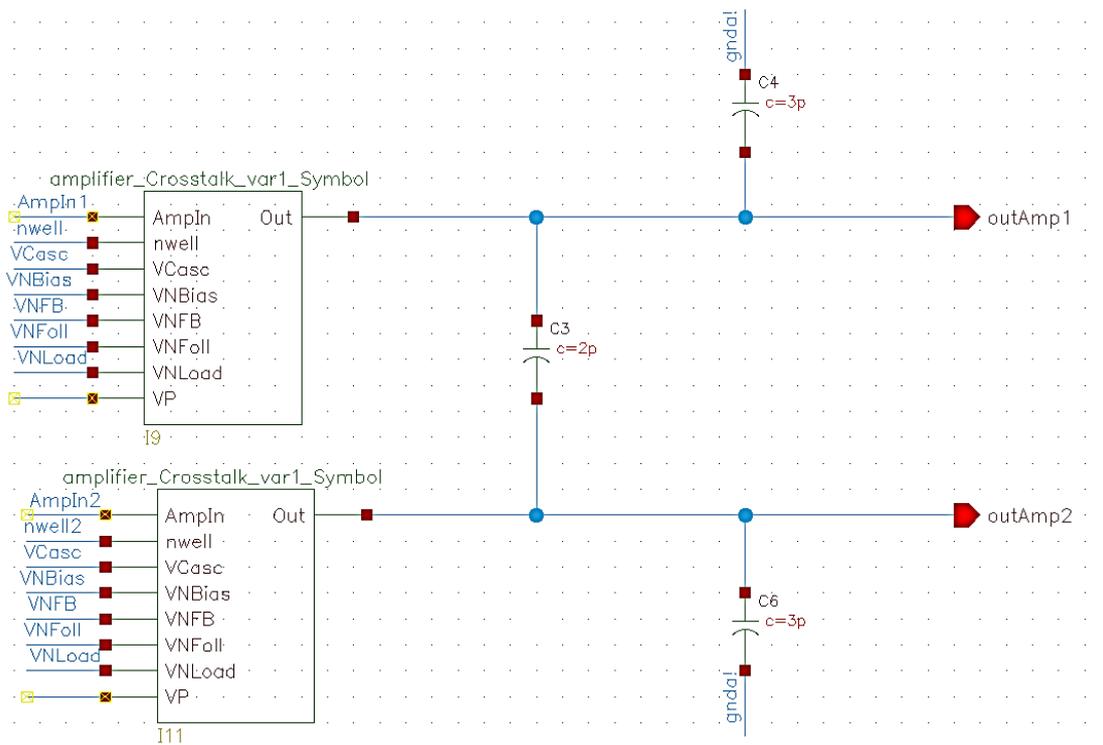


Abbildung 26 Schaltplan Amplifier Crosstalk

Abbildung 26 zeigt den Aufbau der Testschaltung. Hierfür werden zwei identische Verstärker benutzt. Die Kopplung zwischen den Ausgangsleitungen wird durch eine Kapazität mit dem Wert von 2pF repräsentiert. Zusätzlich ist das Ausgangssignal eines Verstärkers mit einer Kapazität von 3pF mit der Versorgungsspannung verbunden und der andere Verstärker mit

der gleichen Kapazität an Masse. Dies ist eine sehr gute Abbildung des möglichen Crosstalk Effekts zwischen zwei Verstärkern von unterschiedlichen Pixeln.

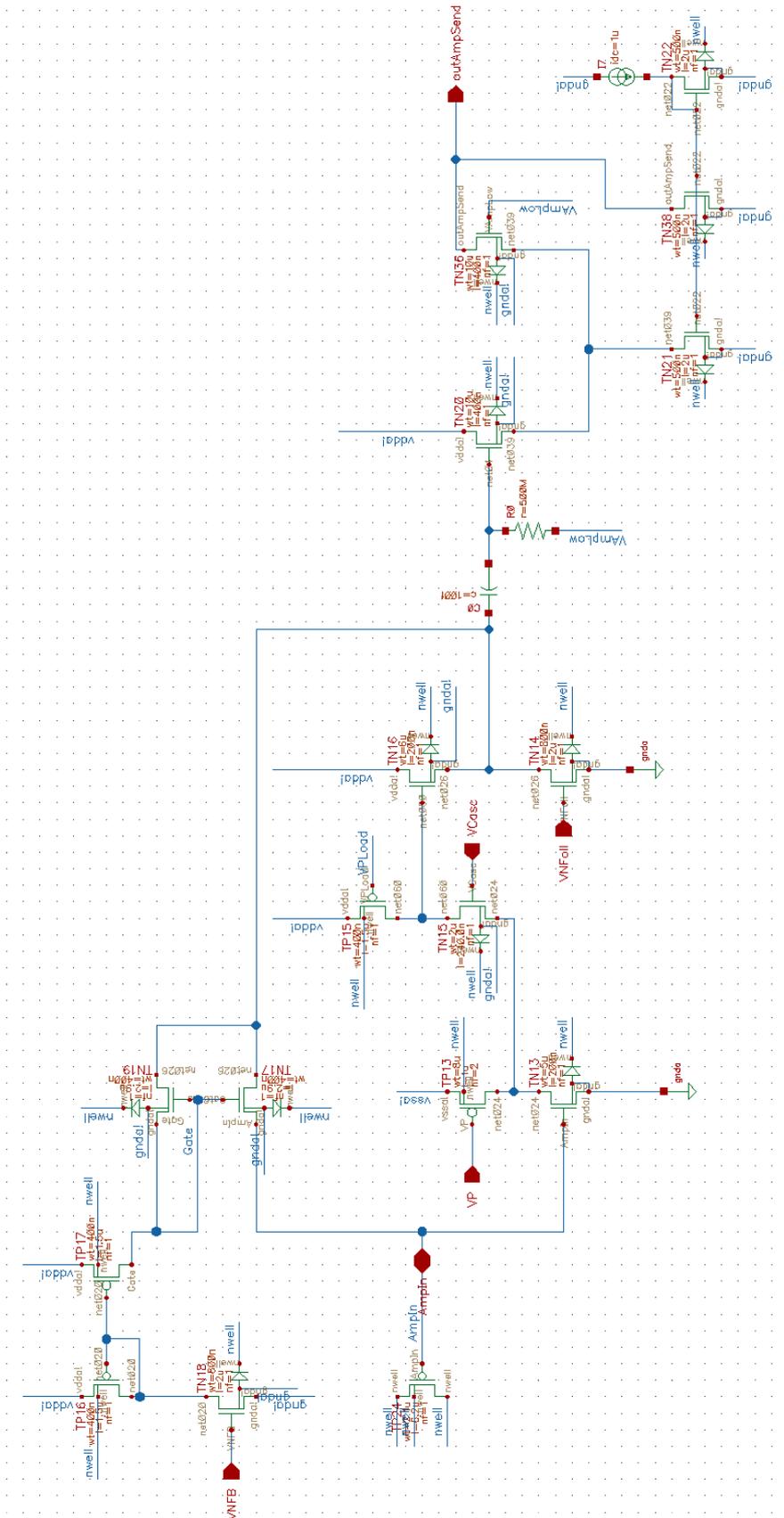


Abbildung 27 Amplifier mit Sendeschaltung

Als nächstes wurde eine ganz andere Möglichkeit zur Behebung des Crosstalk Problems entworfen und simuliert. Die Schaltung wurde der Verständlichkeit halber in zwei Teile getrennt, in den Sendeteil und den Empfängerteil. Die Sendeschaltung mit Verstärker ist zu sehen in Abbildung 27 und die dazugehörige Empfängerschaltung ist in Abbildung 28 gezeigt. Die Idee hierbei ist, dass das Signal als Strom gesendet wird, um einen möglichst geringen Crosstalk zu erreichen. Anschließend transformiert die Empfängerschaltung das Signal wieder zur Spannung zurück.

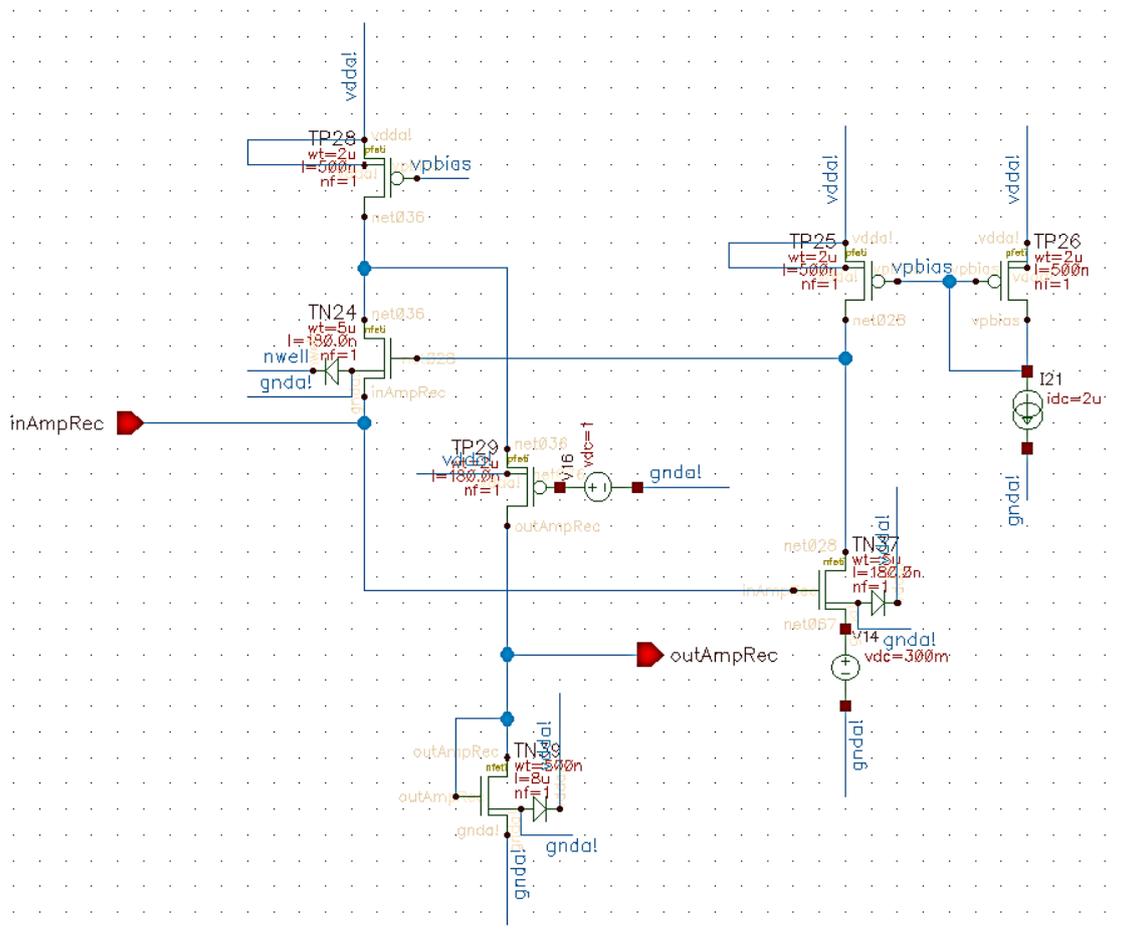


Abbildung 28 Empfängerschaltung

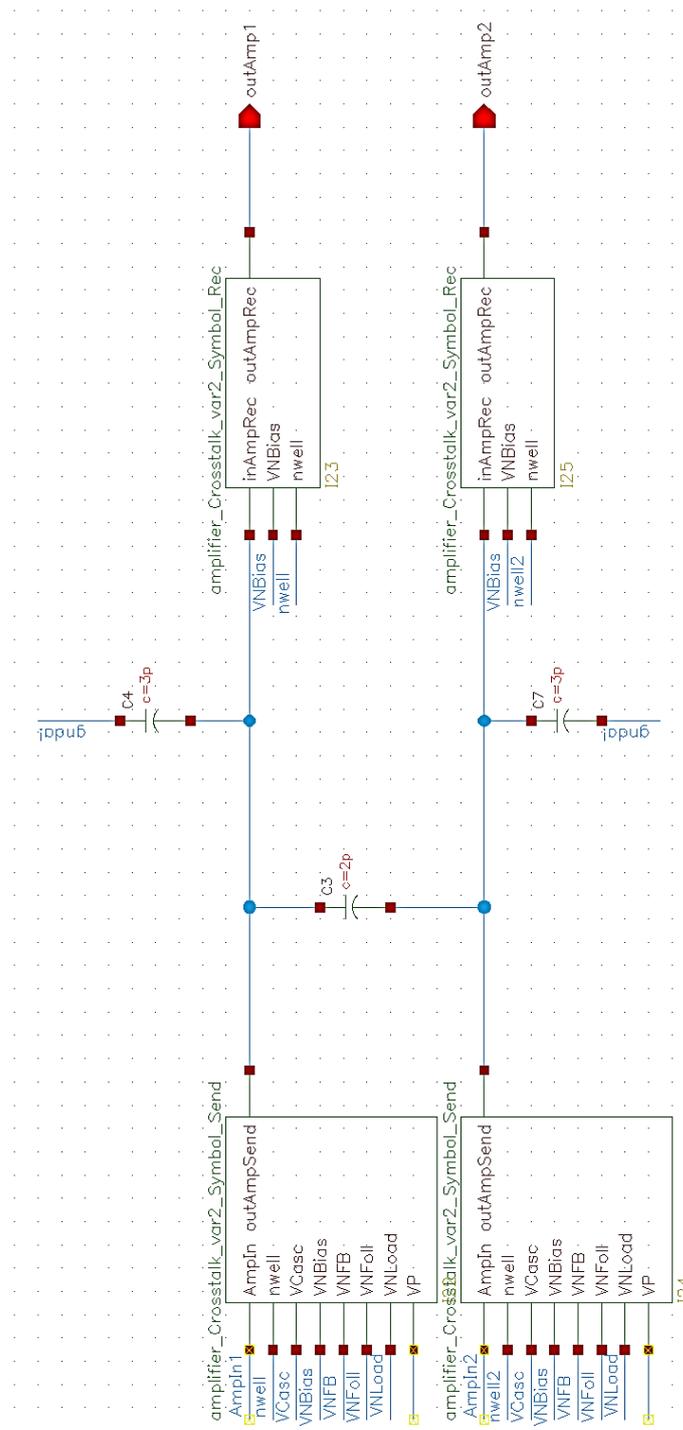


Abbildung 29 Schaltplan Amplifier Crosstalk mit Sende- und Empfangsteil

Nun auch für diese zweigeteilte Verstärkerschaltung der Simulationsaufbau, siehe Abbildung 29. Hier wird nochmals der Gedanke deutlich: im linken Teil befinden sich der Verstärker und der Sender, dann erfolgt die Signalübertragung durch eine Crosstalk gefährdete Leitung, repräsentiert durch die Kapazität zwischen den beiden eigentlich unabhängigen Verstärkern. Nach dieser Übertragung ist der Empfänger geschaltet.

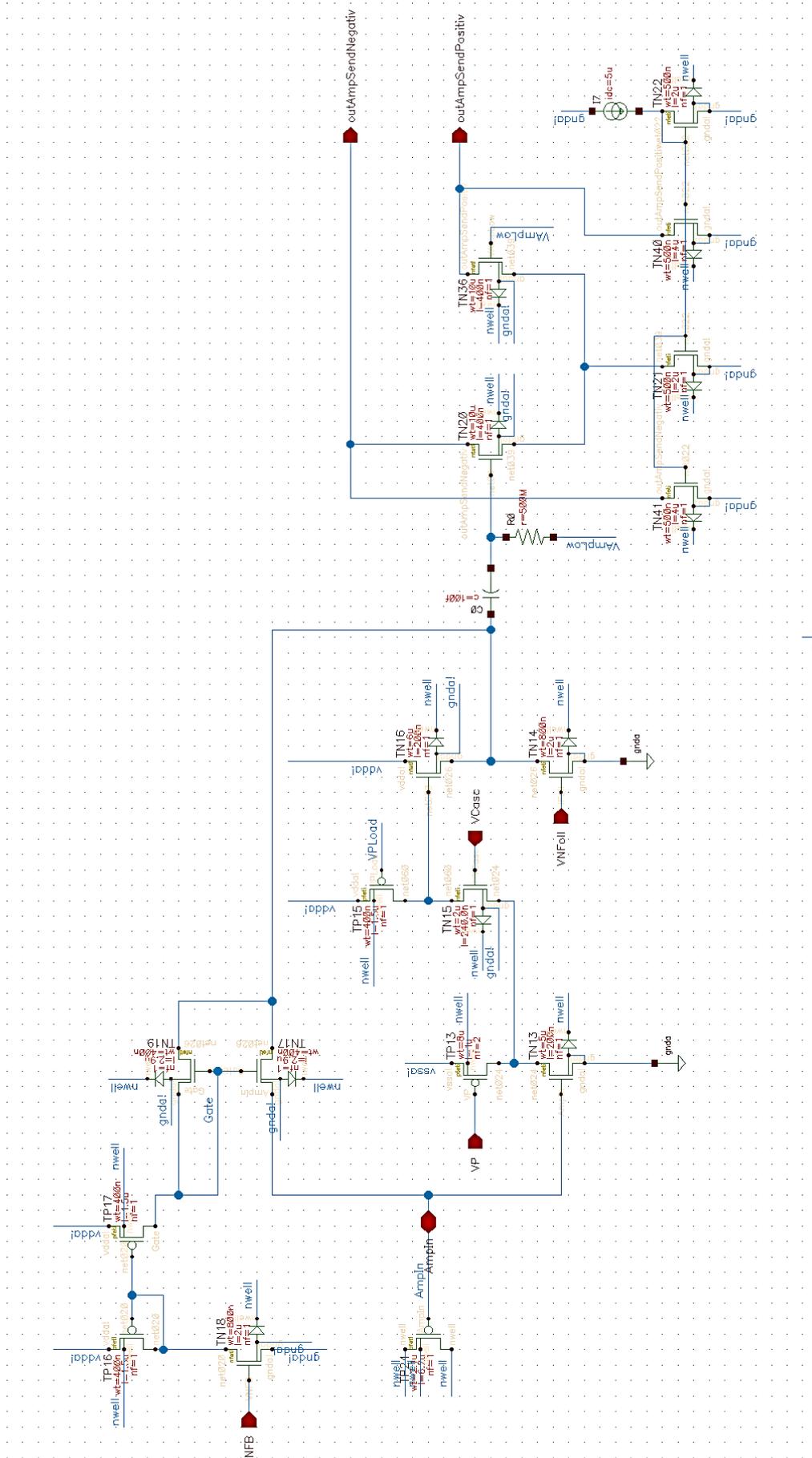


Abbildung 30 Amplifier mit Sendeteil von zwei komplementären Signalen

Eine weitere Möglichkeit der Signalübertragung zwischen Verstärker und Komparator zur Vermeidung von Crosstalk Effekten ist in Abbildung 30 gezeigt. Hier wird aus der vorherigen Variante das Prinzip von Stromübertragung genutzt. Diesmal wird jedoch zusätzlich zu dem positiven Ausgangssignal noch ein weiteres negatives Signal zur Empfängerschaltung übertragen. Die Idee ist, dass diese beiden Ausgangssignale komplementär sind (gespiegelt) und sich dadurch der Crosstalk Effekt aufhebt. Nachteil dieser Möglichkeit ist natürlich der Mehraufwand durch die Übertragung von zwei Signalen, d.h. es werden zwei Leitungen benötigt und dadurch mehr Platz auf dem Chip verbraucht.

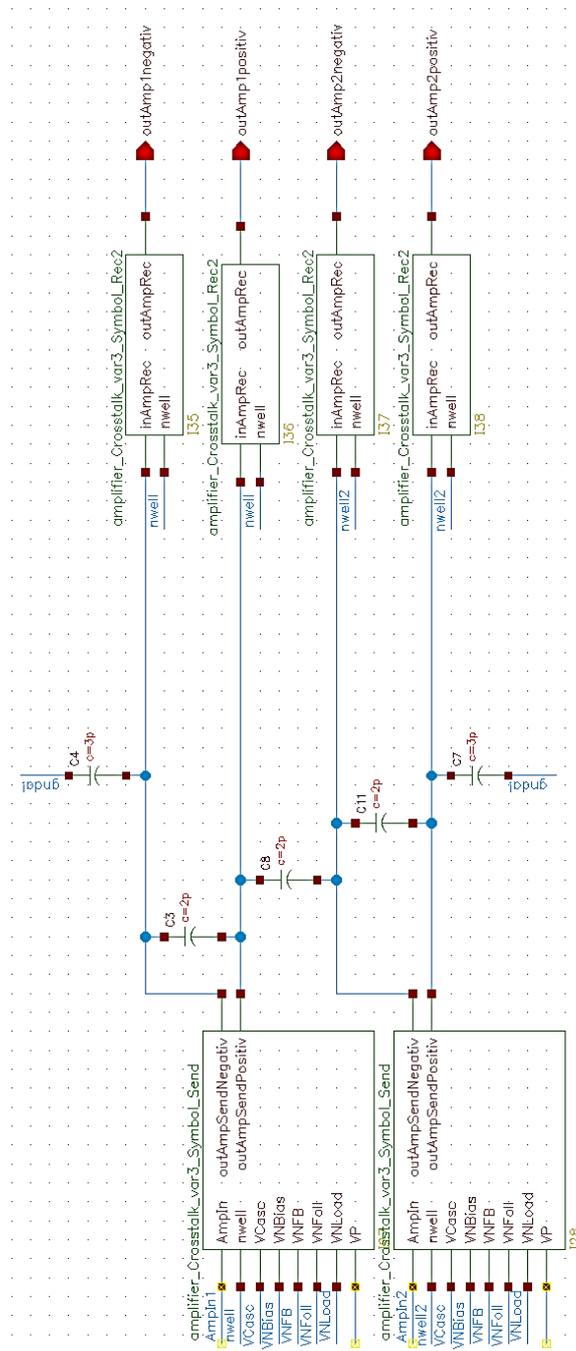


Abbildung 31 Schaltplan Amplifier Crosstalk Sende- und Empfangsteil, Variante mit zwei komplementären Signalen

Ähnlich wie zuvor besteht diese Testschaltung aus dem Verstärker mit Sender, der Kapazität zur Beschreibung der kapazitiven Kopplung der Leitungen und schließlich dem Empfängerteil. Da es sich hier um vier Signale handelt, werden vier Empfänger am Ende benötigt.

### 3.3.3 Simulation

Hier werden für die beiden verschiedenen Schaltungsvarianten die Simulationen vorgestellt, erläutert und bewertet.

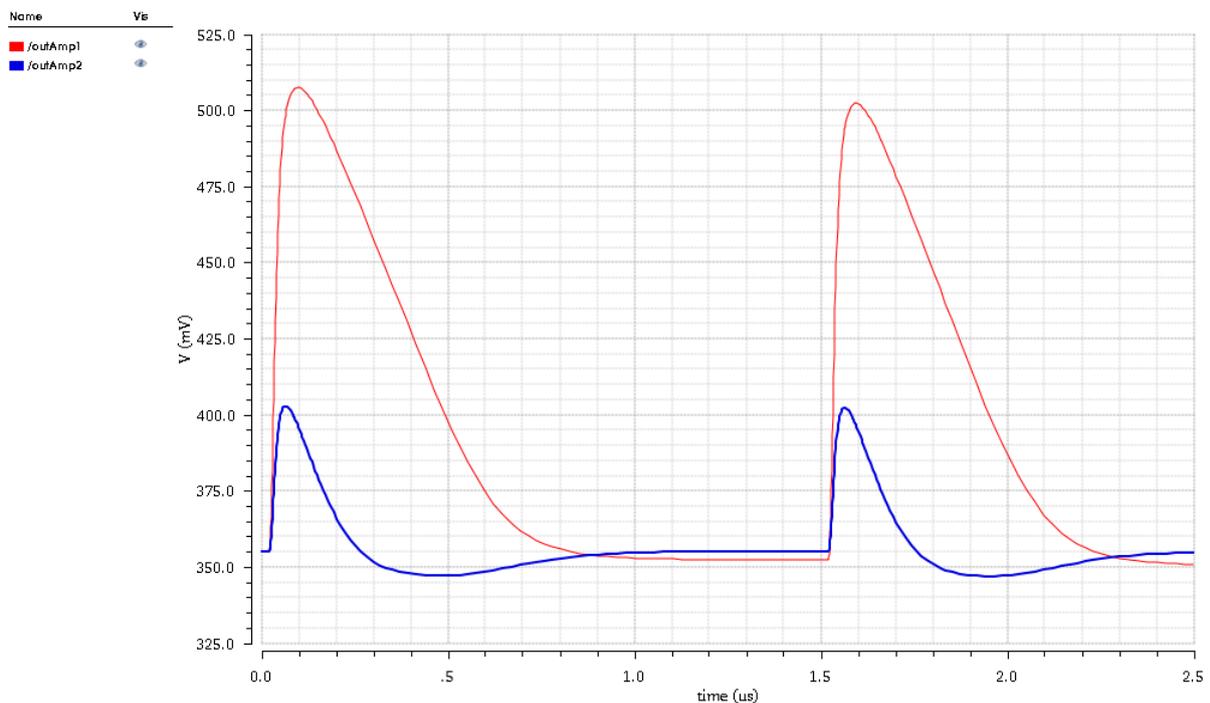
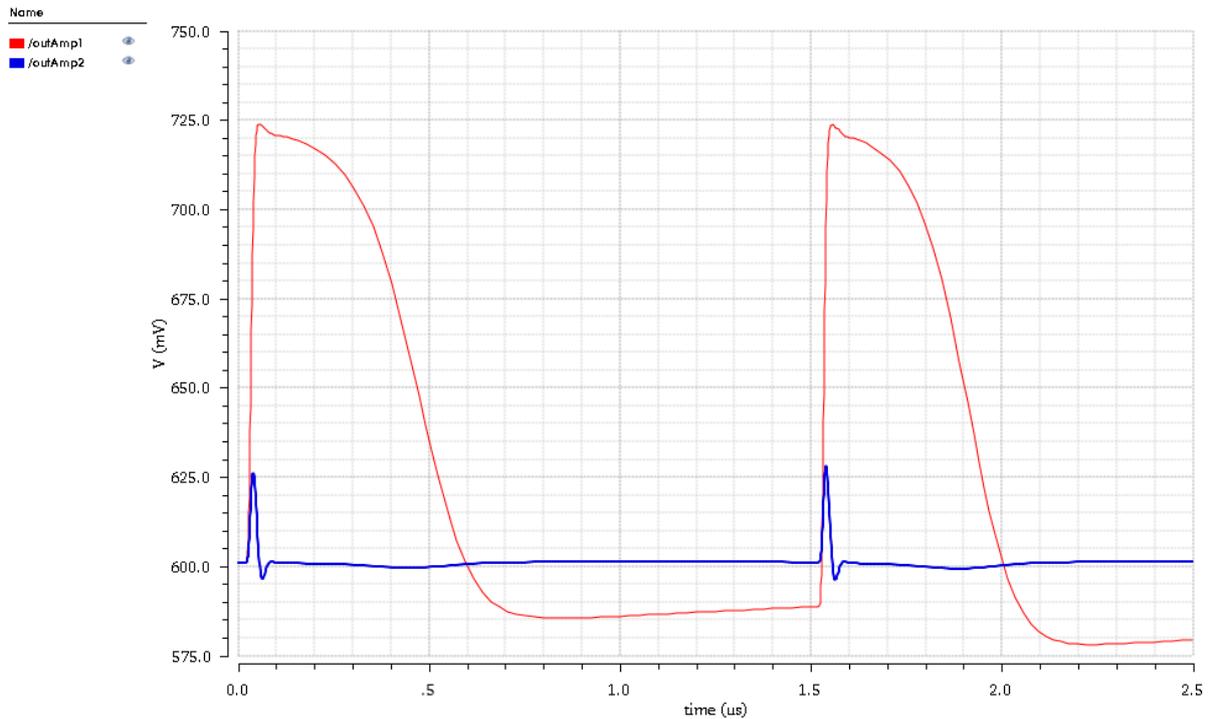


Abbildung 32 Simulation Amplifier Crosstalk

Zunächst der einfachste Fall, die Simulation der Schaltung in Abbildung 26. Dies ist die einfachste Schaltung ohne besondere Anpassungen, um den Crosstalk Effekt zu verhindern. Das rote Signal gehört zu dem Verstärker, der tatsächlich auch ein Signal empfangen hat. Der zu dem blauen Signal gehörende Verstärker hat kein Eingangssignal. Gleichzeitig mit der roten Kurve steigt auch in der danebenliegenden Leitung die Spannung an. Dieser Verlauf zeigt den typischen Verlauf zweier Leitungen, bei denen das Signal der einen über Crosstalk auf die andere Leitung „übertragen“ wird.



**Abbildung 33 Simulation Amplifier Crosstalk Variante mit Stromübertragung**

Nun zu den Ergebnissen der Simulationen der Schaltung aus Abbildung 29 zu sehen in Abbildung 33. Das rote Signal stammt auch hier von dem Verstärker, der auch wirklich ein Signal erhalten hat und weitergibt. Daher ist dieser Verlauf im Vergleich zu der oberen Simulation identisch. Was sich nun jedoch deutlich verändert hat, ist der Spannungsverlauf auf der daneben liegenden Leitung: die blaue Kurve zeigt zwar immer noch eine Beeinflussung durch die rote, allerdings ist diese nun deutlich niedriger, jetzt ungefähr 25mV, davor noch ca. 50mV. Auch zeigt sich diese Beeinflussung nur kurz, etwa 0,1us lang. Zuvor zeigten beide Verläufe, rot und blau, ungefähr die gleiche zeitliche Länge bei einer Signalübertragung. Die Verbesserung des Crosstalk-Verhaltens lässt sich vor allem durch die Trennung in Sende- und Empfängerteil erklären. Allerdings ist nun die maximale Amplitude deutlich kleiner – die Sendeschaltung sättigt.

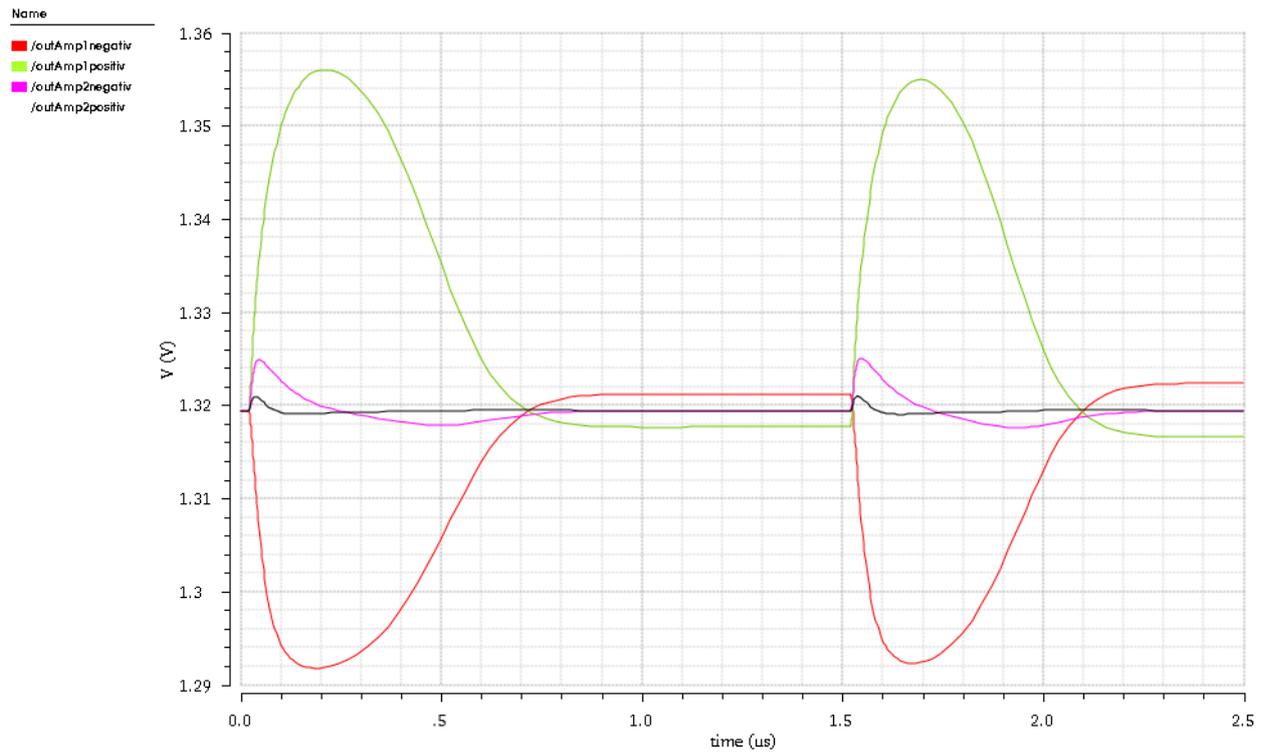


Abbildung 34 Simulation Amplifier mit differentieller Stromübertragung Crosstalk

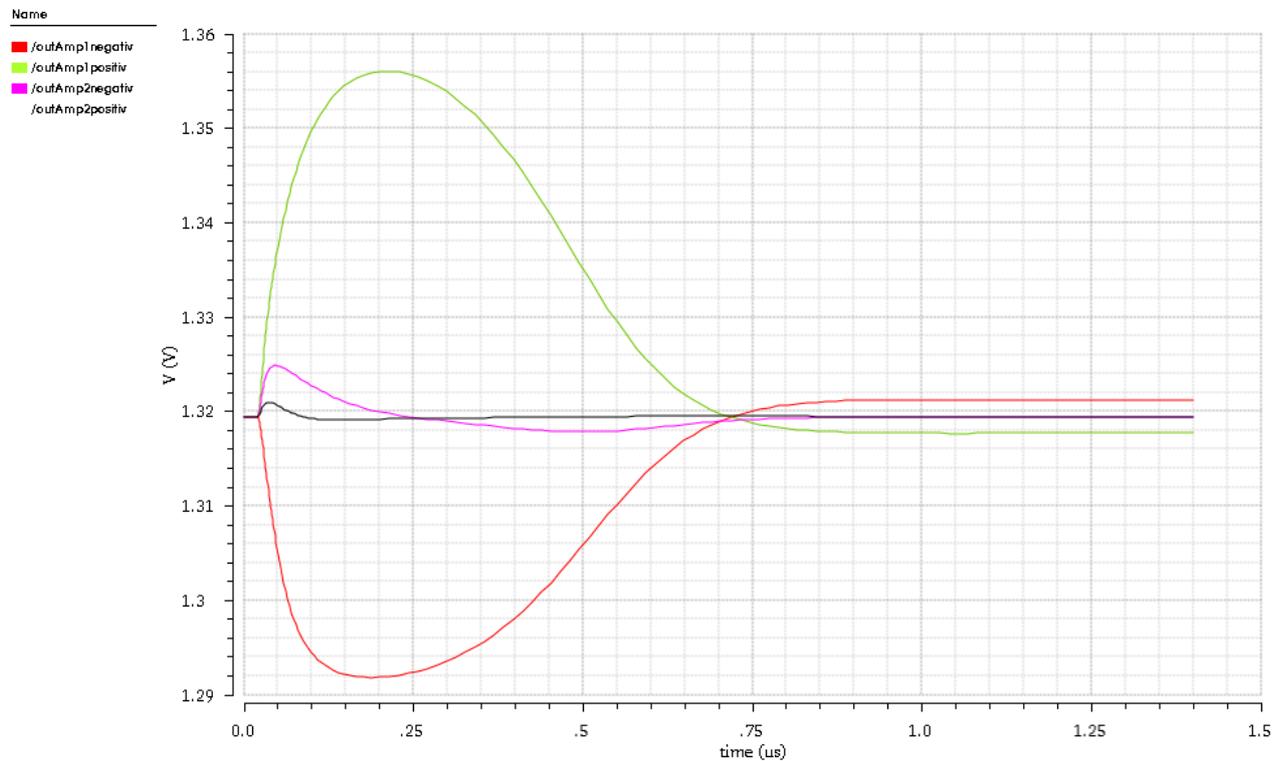


Abbildung 35 Simulation Amplifier mit differentieller Stromübertragung Crosstalk, groß

Als nächstes wurde die Schaltung mit zwei Ausgängen, einmal mit dem ursprünglichen und einmal mit dem gespiegelten Signal, untersucht. Die Ergebnisse der Simulation sind in Abbildung 35 gezeigt. Die grüne Kurve zeigt den positiven Ausgang des Verstärkers, der das

Signal überträgt, die rote ist die dazugehörige gespiegelte Übertragung. Die blaue und lila Kurve zeigen den benachbarten Spannungsverlauf, der durch Crosstalk Effekte mit dem anderen verbunden ist. Zunächst erkennt man sehr schön, dass es sich bei dem zweiten Ausgang wirklich um eine Spiegelung des positiven Ausgangs handelt. Allerdings kommt es trotzdem zum Crosstalk Effekt, wie die beiden anderen Kurven zeigen. Bei dem Verlauf in hellerem lila handelt es sich um den Spannungsverlauf am negativen Ausgang des zweiten Verstärkers, die andere Kurve, dunkler, gehört zu dem positiven Ausgang. Beide zeigen ebenfalls einen Anstieg der Spannung gleichzeitig mit dem Spannungsanstieg des Signals. Die Spannung stieg ungefähr um 5 mV und auf der anderen Leitung um etwa 1 mV Dies ist verglichen mit dem zusätzlichen Aufwand für diese Realisierung, wie in 3.3.1 erklärt, keine Verbesserung im Vergleich zu den zuvor vorgestellten Möglichkeiten. Diese liefern bei weniger Aufwand und damit auch bei niedrigeren Kosten ein akzeptables Ergebnis.

## 3.4 Element: Analog Digital Converter (ADC)

### 3.4.1 Konzept

Der ADC wird durch einen Komparator realisiert. Der Komparator dient dazu, hits, also relevante Signale, von anderen unwichtigen, wie zum Beispiel Rauschen, zu trennen. In der einfachsten Form wird ein Komparator verwendet, dieser vergleicht die Spannung am Eingang mit einer vordefinierten Referenzspannung. Übersteigt die Spannung diesen Referenzwert, so schaltet der Ausgang auf high, in allen anderen Fällen bleibt er bei low. Damit fungiert der Komparator hier als Analog-Digital-Wandler, da er die analogen Signale in digitale umwandelt, die nur noch aus high oder low bestehen.

Im Folgenden wird diese Schaltung vorgestellt und optimiert. Die große Weiterentwicklung des ADC wird in 3.5 erläutert.

### 3.4.2 Schaltplan

In Abbildung 36 sieht man die ursprüngliche Form des für Mu3e verwendeten Komparators.

Im Vergleich zur älteren Version des Komparators wird nun in der neuen Version in Abbildung 37 das Signal nicht über die gleiche Seite in die Schaltung eingespeist, wo später der Ausgang liegt, sondern über die gegenüberliegende. Es gibt also eine bessere räumliche Trennung von Ausgangssignal und Eingangssignal. Ziel dieser Änderung ist eine bessere Entkopplung der Signale. Dies wirkt unerwünschten Wechselwirkungen entgegen, die die Funktion der Schaltung negativ beeinflussen.

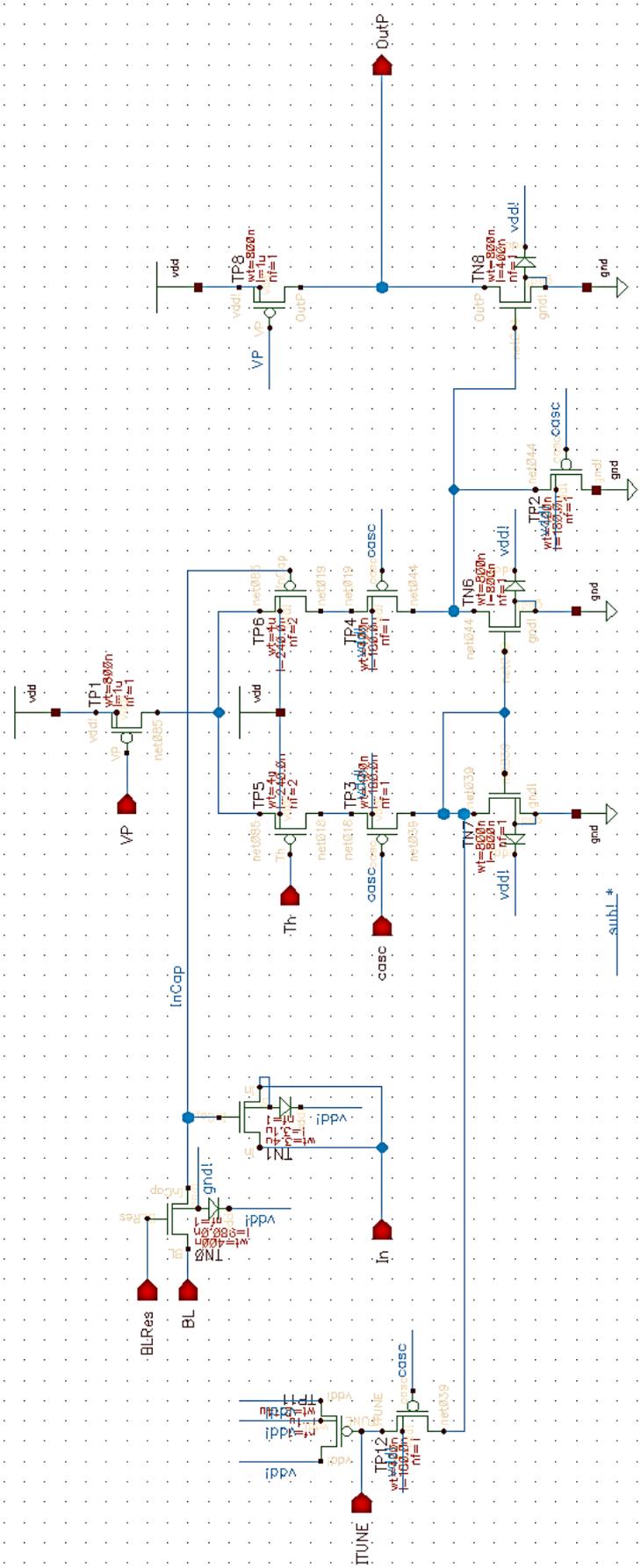


Abbildung 36 Vorgängerversion des Komparators

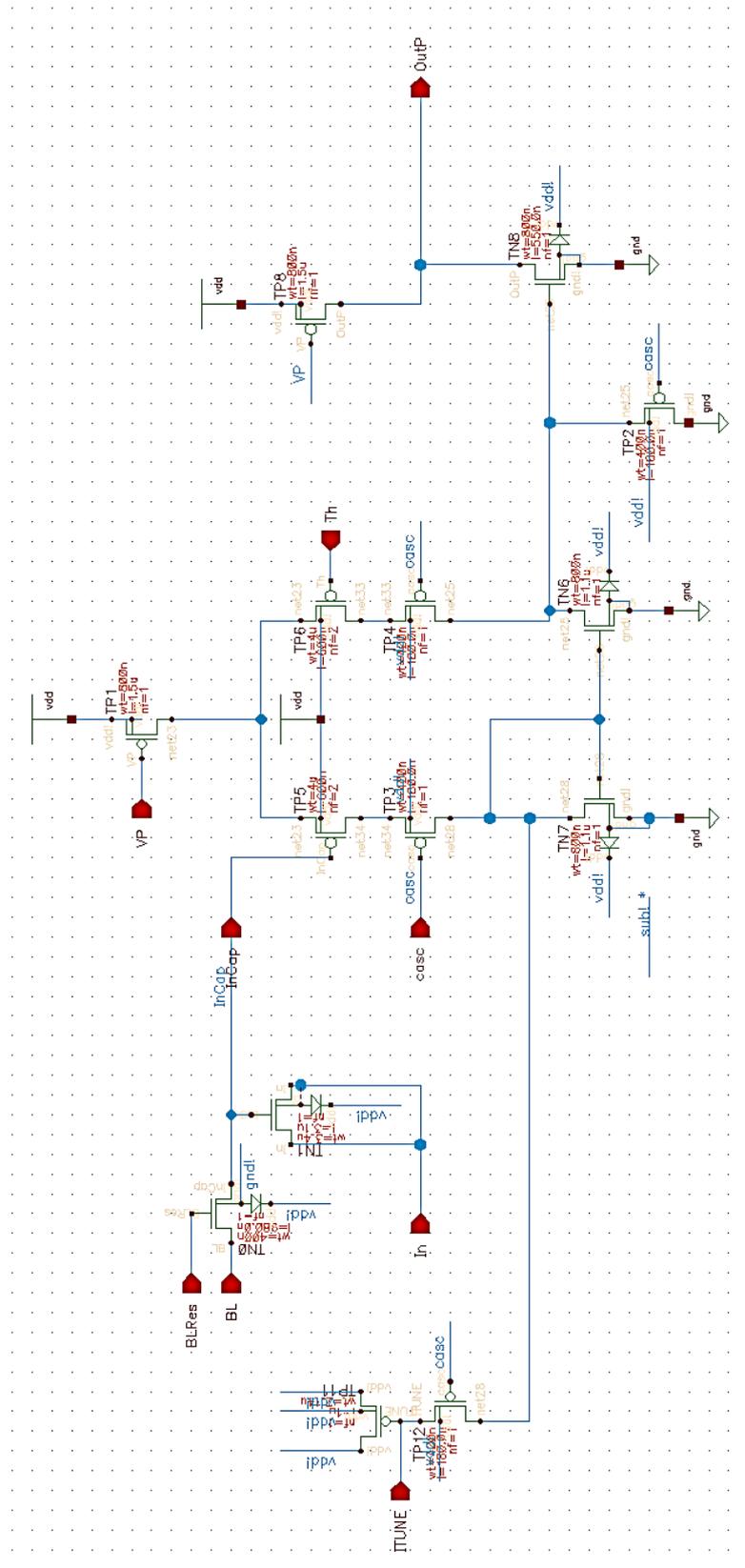


Abbildung 37 Optimierung des Komparators, Teil 1

In der nächsten Weiterentwicklung wurde auf die interne AC Kopplung verzichtet. Diese kann so bei Bedarf nach außen, vor dieses Element, verlegt werden. Jedoch kann so auch einfacher entschieden werden, ob diese zu einer Verbesserung der Gesamtschaltung beiträgt

oder nicht. Zudem wurde hier auf den Transistor TP2 verzichtet, da dieser für die erwünschte Funktion des Komparators nicht notwendig ist. Jede Einsparung an Transistoren bedeutet einen gewissen Vorteil, da so weniger Chipfläche benötigt wird und mehr Platz für andere Funktionen bleibt. Dabei muss man bedenken, dass der Komparator mehrere hundert Mal auf dem Chip platziert sein wird.

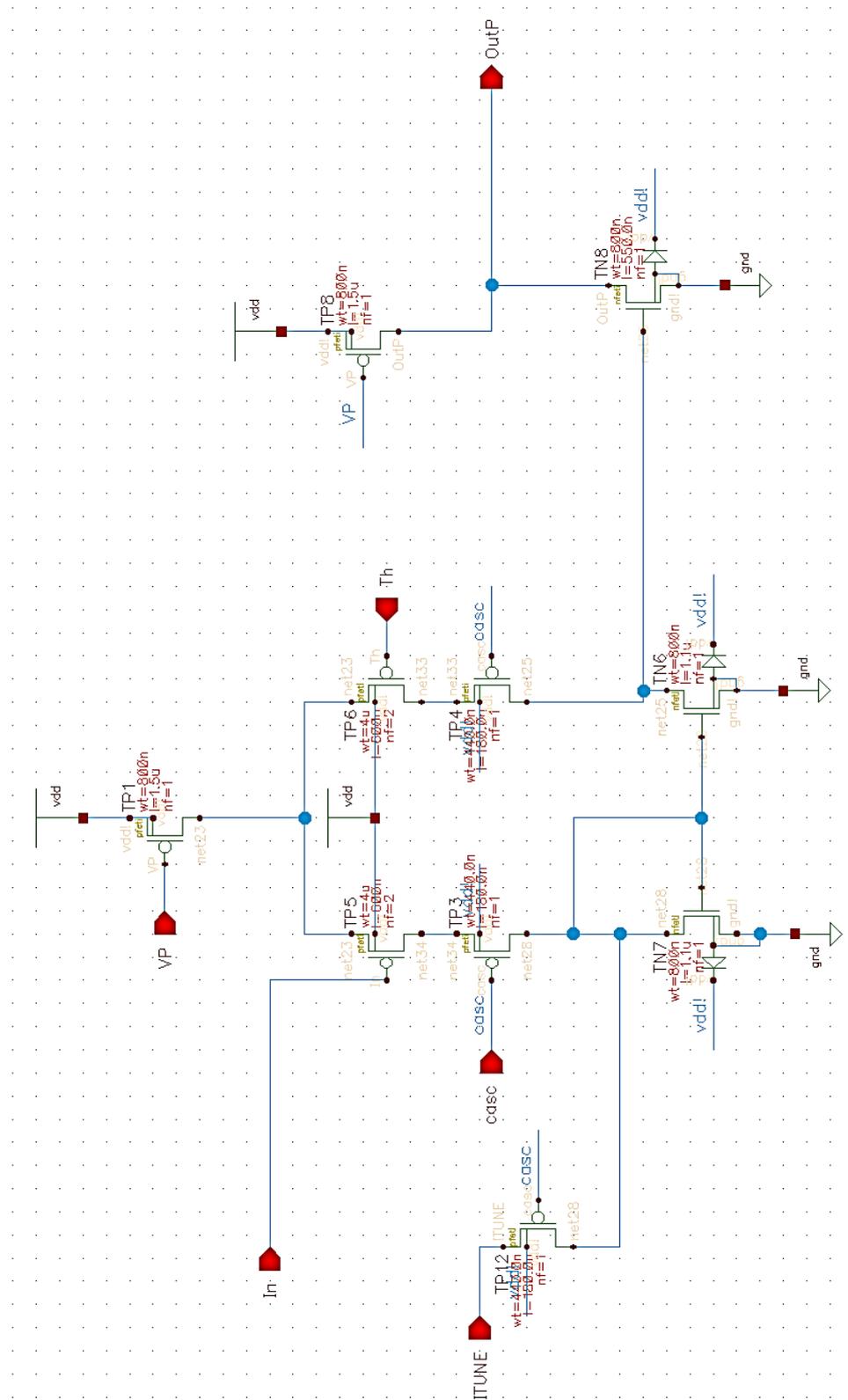


Abbildung 38 Optimierung des Komparators, Teil 2

Als nächstes muss die benötigte Schwellspannung für die folgenden Simulationen bestimmt werden. Hier wurde mithilfe von Simulationen eine günstige Schwellspannung bestimmt. Diese kann später selbstverständlich von außen angelegt werden und dadurch verändert werden. Trotzdem ist es für die folgenden Simulationen wichtig, einen identischen Wert für die Schwellspannung zu nehmen, um die Ergebnisse verschiedener Simulationen besser vergleichen zu können.

### 3.4.3 Simulation

#### 3.4.3.1 Ideale Simulation

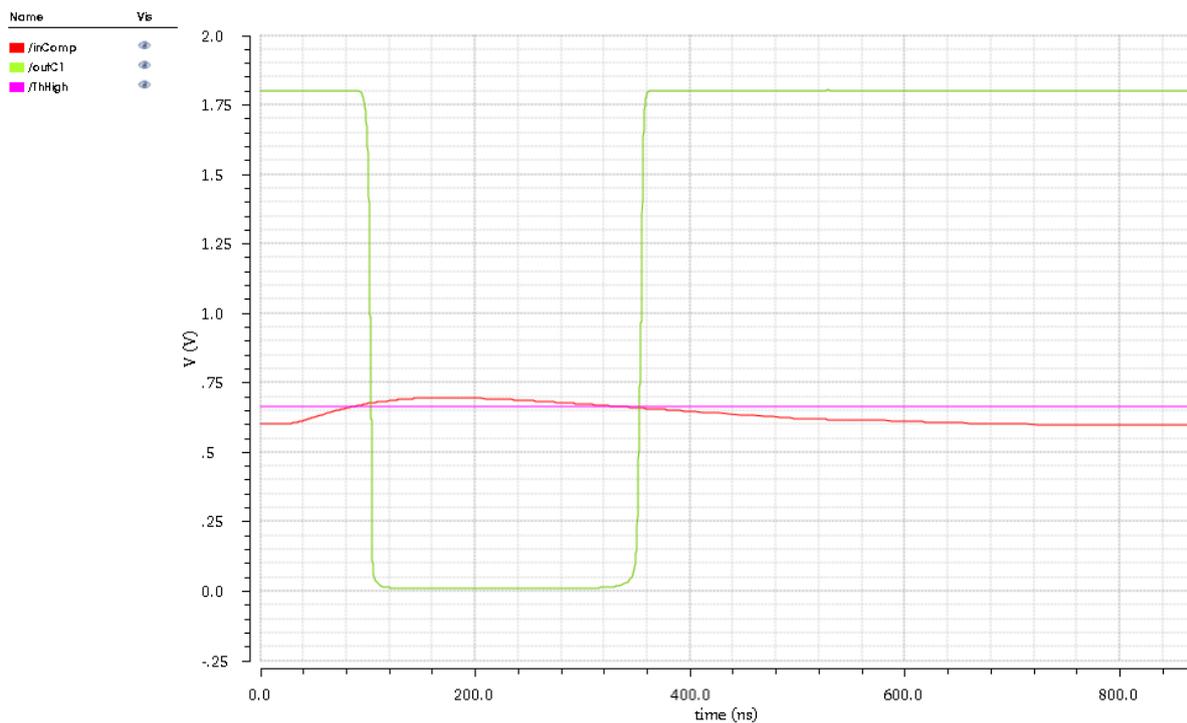


Abbildung 39 Komparator mit einer Schwelle

Hier ist das Ergebnis der Simulation aus der Abbildung 38 zu sehen. Hier wurden alle Veränderungen des Schaltplans durchgeführt und die Simulation zeigt das gewünschte Ergebnis. Um das Ergebnis besser erklären zu können, eine Vergrößerung der oberen Grafik.

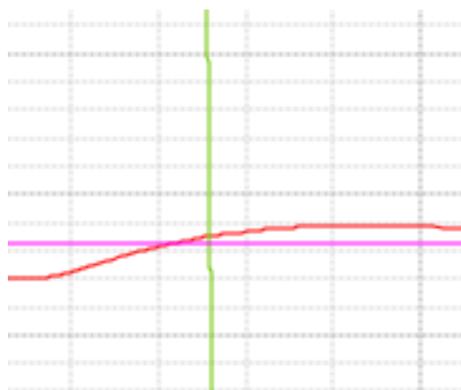
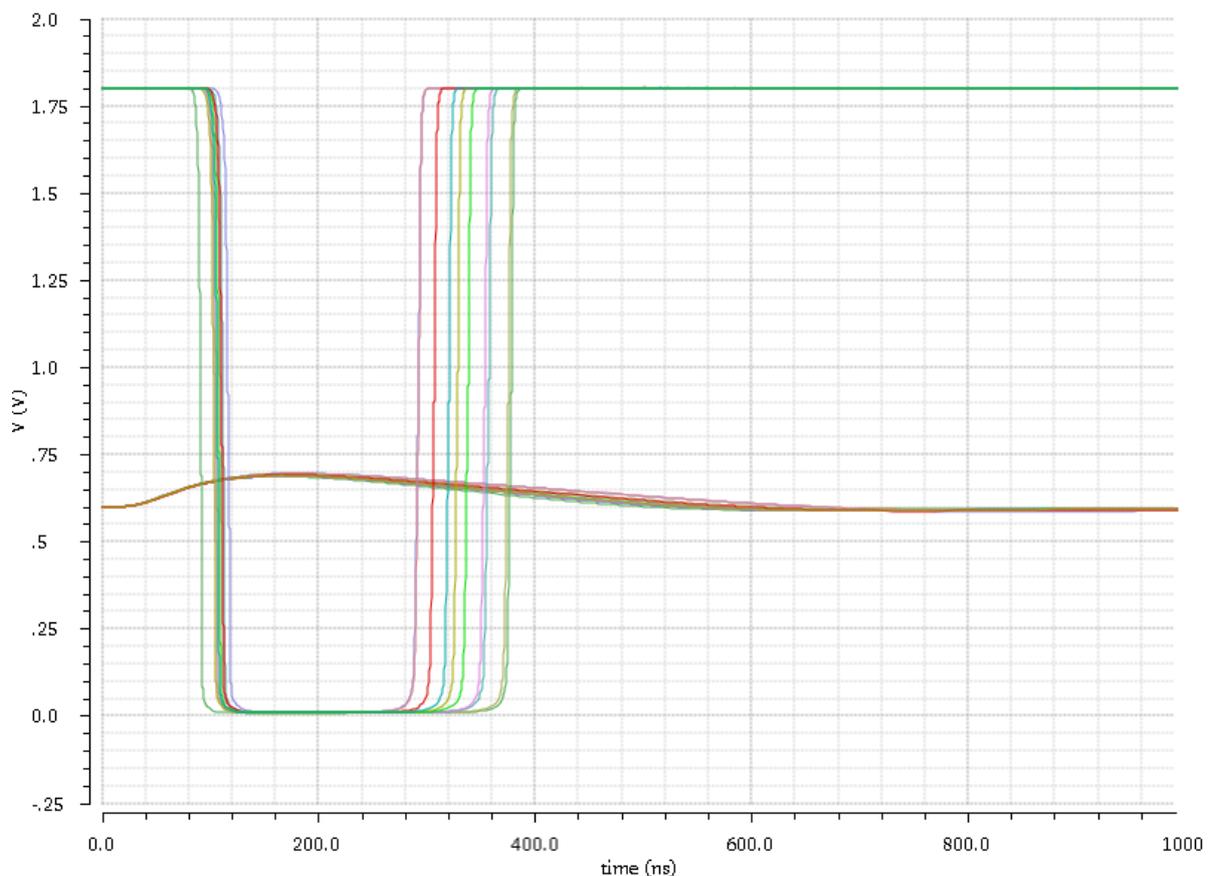


Abbildung 40 Komparator mit einer Schwelle, vergrößert

Die pinkfarbene Linie ist die konstante Schwellspannung, die rote zeigt den Verlauf des Eingangssignals. Sobald die Schwelle vom Signal überschritten wird, schaltet der Komparator und der Ausgang, grüne Linie, ändert sich. Der Komparator schaltet beim Überschreiten der Schwelle von high auf low. Die Qualität eines Komparators zeigt sich daran, dass er sofort nach dem Überschreiten schaltet, also die Ausgangsspannung beginnt zu fallen, und daran, dass der Endwert, hier low, in möglichst kurzer Zeit erreicht wird. Zusammenfassend sind also steile Schaltflanken erwünscht. Dies ist in der hier entwickelten Schaltung der Fall. Der Komparator benötigt unter 15ns zum Schalten und auch der Rückschaltevorgang auf high erfolgt in der gleichen kurzen Zeit.

### 3.4.3.2 Monte Carlo Verfahren

Wie auch für den Verstärker wurden mehrere Monte Carlo Simulationen für den Komparator durchgeführt, um die Auswirkung des Mismatch der Transistoren genau zu untersuchen. Beispielhaft ist in Abbildung 41 eine Monte Carlo Simulation gezeigt, zur Übersichtlichkeit mit unter 20 Werten. Es wurden für jedes entworfene Elemente auch Simulationen mit über 300 Werten durchgeführt. In dem gezeigten Simulationsergebnis ist sehr deutlich, dass die erste Flanke, wenn der Komparator schaltet, von high auf low, nur schwach durch Mismatch verfälscht wird. Dies ist sehr positiv, da für die AD-Konvertierung genau diese Flanke entscheidend ist. Die spätere Flanke spielt hierfür keine Rolle mehr, daher ist es unkritisch, dass diese stärker durch Mismatch beeinflusst wird.



**Abbildung 41 Monte Carlo Simulation Komparator**

### 3.4.3.3 Simulation mit Rauschen

In Abbildung 42 ist die gleiche Simulation wie zuvor gezeigt, allerdings wurde nun zusätzlich das Rauschen berücksichtigt. Man erkennt, dass die Schaltung recht rauscharm ist und der Umschaltvorgang vom Rauschen nicht beeinflusst wird. Dies ist ebenfalls ein Vorteil dieser Schaltung. Auch deutlich ist, dass es auch bei einem sehr kleinen Anfangsstrom von 20 nA eine eindeutige Unterscheidung zwischen Rauschen und Signal gibt, die Schwellspannung wurde also zuvor korrekt berechnet.

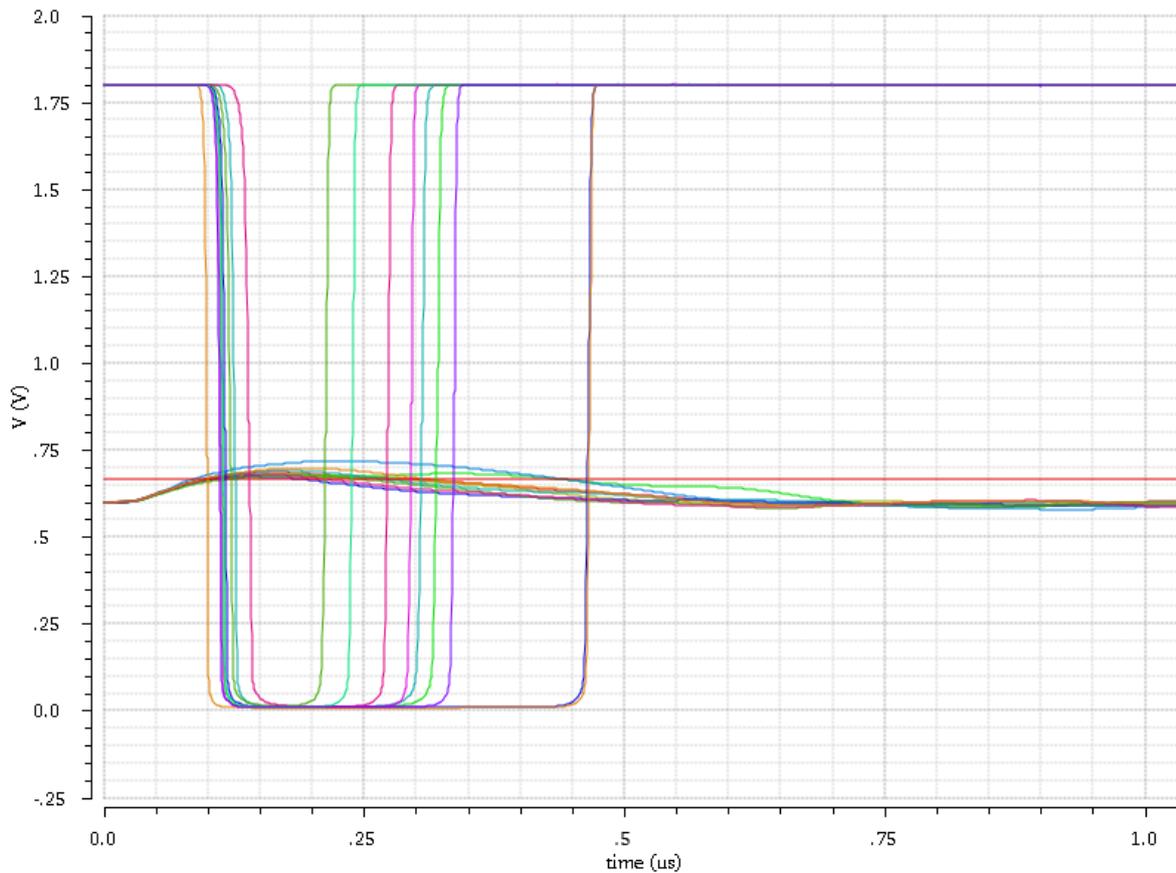


Abbildung 42 Rauschsimulation Komparator

## 3.5 Zwei Modi für den ADC: zwei Schwellen-Modus und Amplitudenmessung-Modus

### 3.5.1 Konzept

Bisher wurde für den Pixelchip des Mu3e Projekts nur ein Komparator mit einer Schwelle für jeden Pixel verwendet. Dies birgt jedoch ein Problem: den Timewalk.

Der Zeitpunkt des hits, genannt TimeStamp, wird aufgezeichnet, sobald das Signal die Schwelle überschreitet. Das große Problem ist hier, dass der Zeitpunkt zwischen Beginn des Signals und Überschreiten der Schwelle je nach Höhe der Eingangsspannung variiert. Dies führt zu einer Verfälschung des TimeStamp Wertes, dem sogenannten Timewalk. Abhängig von der Größe des Eingangsstroms gibt es also eine unterschiedlich große Verschiebung zwischen dem Zeitpunkt des hits, bei dem das Signal beginnt zu steigen, und dem Zeitpunkt der Überschreitung der Schwelle, der als TimeStamp aufgezeichnet wird. Je höher die Schwelle, desto größer dieser unerwünschte Effekt. Daher kann die Verschiebung nicht einfach nachträglich heraus gerechnet werden.

Um dieses Problem zu beheben, wurde ein neues Konzept entwickelt, beschrieben wird dies in 3.5.3.1. Außerdem wurde noch ein weiteres neues Konzept entwickelt: eine Amplitudenmessung, siehe 3.5.3.2. Diese beiden Konzepte sollen später gleichzeitig auf dem Chip als zwei verschiedene Modi realisiert werden. Zwischen den beiden Modi kann man dann später einfach umschalten.

Für beide Modi ist es notwendig, dass zwei Komparatoren statt bisher nur ein Komparator für jedes Pixelelement verwendet werden. Daher ist es später in 5.3 erforderlich, ein besonders kompaktes Layout für die Komparatoren zu entwickeln.

#### 3.5.1.1 Zwei Schwellspannungen

Zunächst zu dem Modus mit zwei Schwellspannungen. Durch die Verwendung von zwei Komparatoren ist es möglich, zwei verschiedene Schwellspannungen anzulegen. Eine der Schwellspannungen wird, wie in 3.4.3, gewählt. Diese hat die Aufgabe, sicherzustellen, dass es sich bei der Spannung um ein Signal und nicht nur um Rauschen handelt. Daher darf diese nicht zu niedrig gewählt werden. Selbstverständlich darf sie jedoch auch nicht zu hoch gewählt werden, da sonst Treffer nicht als solche erkannt werden. Daher muss sowohl das Rauschen als auch die minimale Signalspannung berücksichtigt werden.

Die andere Schwellspannung kann nun jedoch deutlich niedriger gewählt werden. Sobald die Eingangsspannung diese Schwelle überschreitet, was deutlich früher der Fall sein wird, als die andere höhere Schwelle überschritten wird, wird die zugehörige TimeStamp aufgezeichnet. Dies vermindert den Timewalk Effekt erheblich, da nun der Zeitpunkt viel früher gespeichert werden kann. Die Verschiebung, wann verschieden hohe Eingangsströme die Schwelle überschreiten, ist bei einer derart niedrigen Schwelle deutlich kleiner. Dies wird veranschaulicht durch die Simulation in 3.5.3.1. Übersteigt das Signal anschließend auch die höhere Schwelle, so handelt es sich sicher um einen hit, der Komparator schaltet auf high, das hit Signal wird erzeugt und trotzdem die frühe TimeStamp des ersten Komparators

verwendet. Wird die zweite Schwelle nicht überschritten, so wird kein hit Signal weitergegeben und der TimeStamp Wert einfach bei der nächsten Spannung, die die niedrigere Schwelle überschreitet, überschrieben.

Dieser Modus bewirkt damit eine große Verbesserung des Timewalk Problems und behält trotzdem die Eigenschaften der Grundschaltung in Bezug auf die Funktion.

### 3.5.1.2 Rampe

Der nächste Modus erlaubt nun zusätzlich eine Amplitudenmessung. Ein Komparator bleibt unverändert bei oben erklärten Modus: der Komparator mit der höheren Schwelle. Dieser dient weiterhin zur Detektion eines Signals durch Unterscheidung des Signals von Nebeneffekten wie Rauschen. Der andere Komparator hat nun jedoch nicht mehr wie zuvor eine konstante Schwellspannung, sondern eine linear steigende Schwellspannung. Je nach Höhe des Eingangssignals unterscheidet sich nun der Verlauf der Ausgangsspannung des Komparators. Je niedriger das Eingangssignal, desto früher schaltet der Komparator zurück, und je höher das Eingangssignal, desto später schaltet der Komparator wieder in den Ausgangszustand. Dies funktioniert am besten mit kleinen Eingangsspannungen, diese liegen weiter auseinander als gleich weit voneinander entfernte höhere Eingangssignale. Mit diesem Modus lässt sich somit eine weitere Information erhalten als nur die, ob es einen Treffer gab: nun gewinnt man eine Aussage über die Höhe des durch den Treffer detektierten Signals. Dies erlaubt auch Rückschlüsse auf die Höhe des erzeugten Stromes und damit auf die Energie des Teilchens.

### 3.5.2 Schaltplan

Hier wird der neu entworfene Schaltplan vorgestellt, zu sehen in Abbildung 43. Gleich auf den ersten Blick erkennt man die beiden Komparatoren. Beide sind an das gleiche Eingangssignal angeschlossen. Vor den Komparatoren befindet sich eine AC Kopplung. Die AC Kopplung wird auch als kapazitive Kopplung bezeichnet und dient dazu, die DC Anteile des Signals zu filtern. So ist es möglich, kleinere AC-Signaländerungen besser zu erkennen. Die AC Kopplung wurde außerhalb des Elements des Komparators gelegt, so können sich beide Komparatoren das Ausgangssignal der AC Kopplung teilen. Der im Schaltplan unten dargestellte Komparator ist der mit der höheren Schwellspannung. Die Schaltung vor dem oberen Komparator ermöglicht das Umschalten zwischen den beiden Modi und realisiert die linear ansteigende Schwellspannung. Die steigende Schwellspannung wird mithilfe eines Kondensators realisiert, der geladen wird. Dieser wird jedoch durch einen Transistor (TN17, zu finden rechts neben dem Transistor, an den RampEnB angeschlossen ist) ersetzt, daher ist im Schaltplan kein Kondensator direkt zu finden. Die Höhe der niedrigen Schwellspannung wird durch den ganz links zu sehenden Input ermöglicht („Th low“). Mit Hilfe des Inputs „VP Rampe“ kann im Modus der Amplitudenmessung die Steigung der Geraden eingestellt werden. Es können daher diese beiden wichtigen Größen später bei den Messungen optimal eingestellt werden und diese Entscheidung muss noch nicht endgültig beim Design getroffen werden.

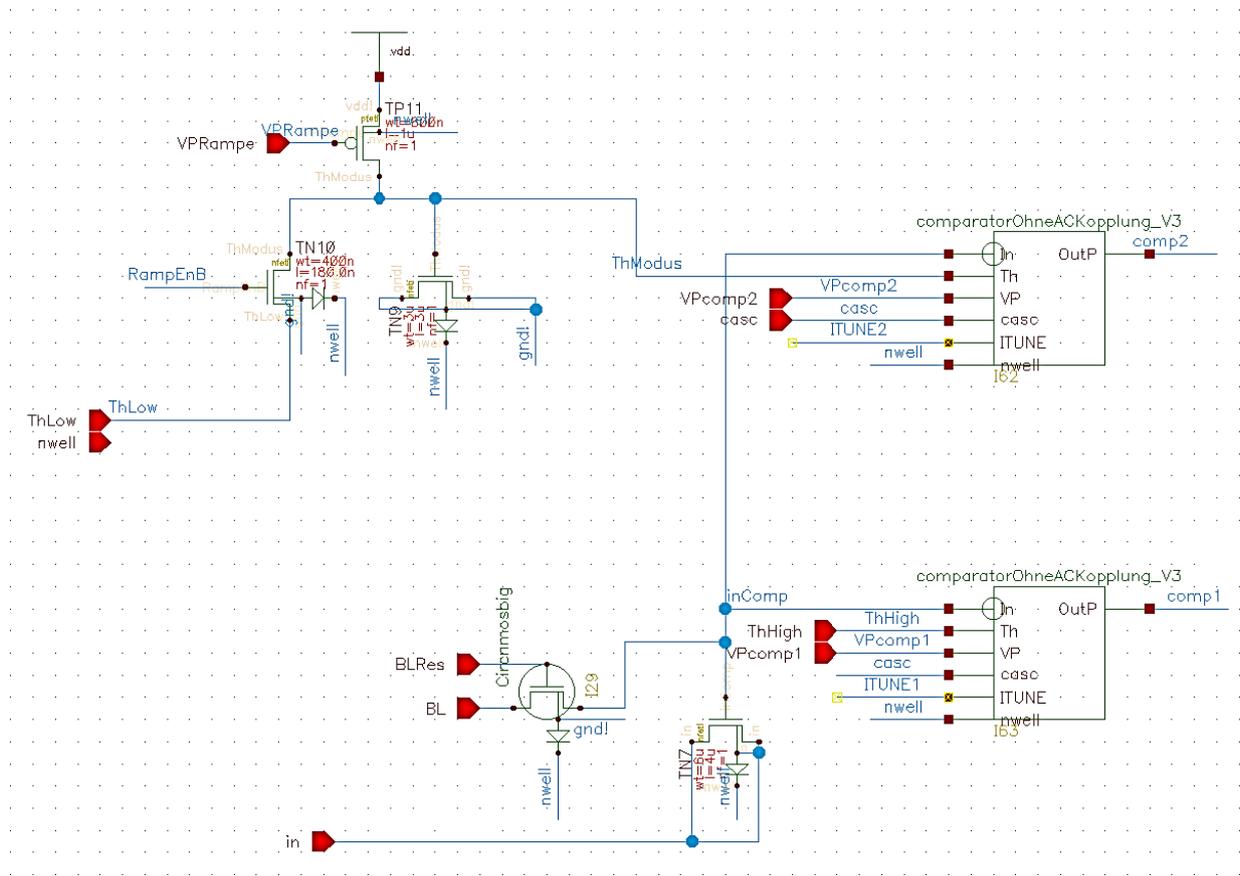


Abbildung 43 Schaltplan, zwei Modi für ADC

### 3.5.3 Simulation

#### 3.5.3.1 Simulation Modus I: zwei Schwellen

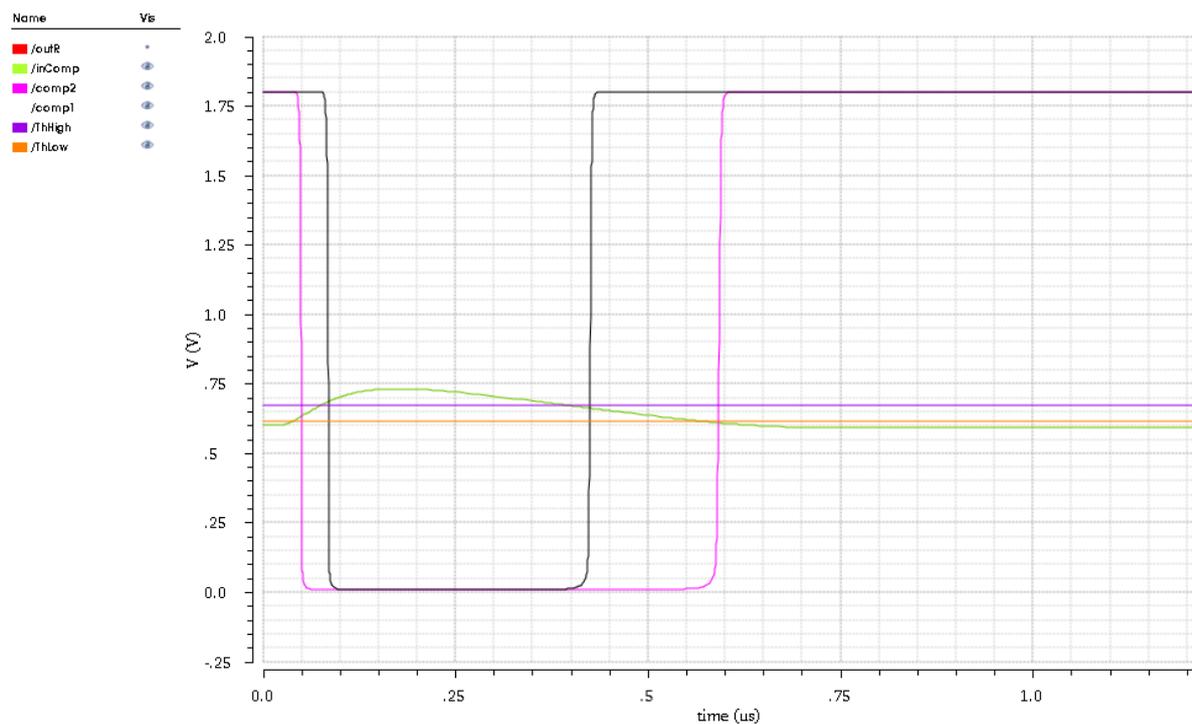


Abbildung 44 Simulation Komparator zwei Schwellen

Hier sieht man das Ergebnis der Simulation des ersten Modus bei einem Eingangsstrom von 20nA. Die lila Gerade zeigt die obere Schwelle, die durch einen Strom von 70nA erzeugt wird und damit bei ~ 0,655V liegt, die untere liegt bei ~ 0,62V, erzeugt durch einen Strom von 15nA. Deutlich erkennt man, dass der im Schaltplan obere Komparator (pink) sehr viel früher schaltet als der andere.

Um die Verbesserung des Timewalk Problems zu zeigen, wird auch eine Simulation für verschiedene Eingangssignale gezeigt, Abbildung 45. Die verschiedenen Zeitpunkte, zu denen die niedrige Schwelle überschritten wird (grüne Kurven), liegen deutlich näher zusammen als die Zeitpunkte, zu denen die hohe Schwelle passiert wird (rote Kurven). Dies bestätigt die zuvor getroffene Designentscheidung.

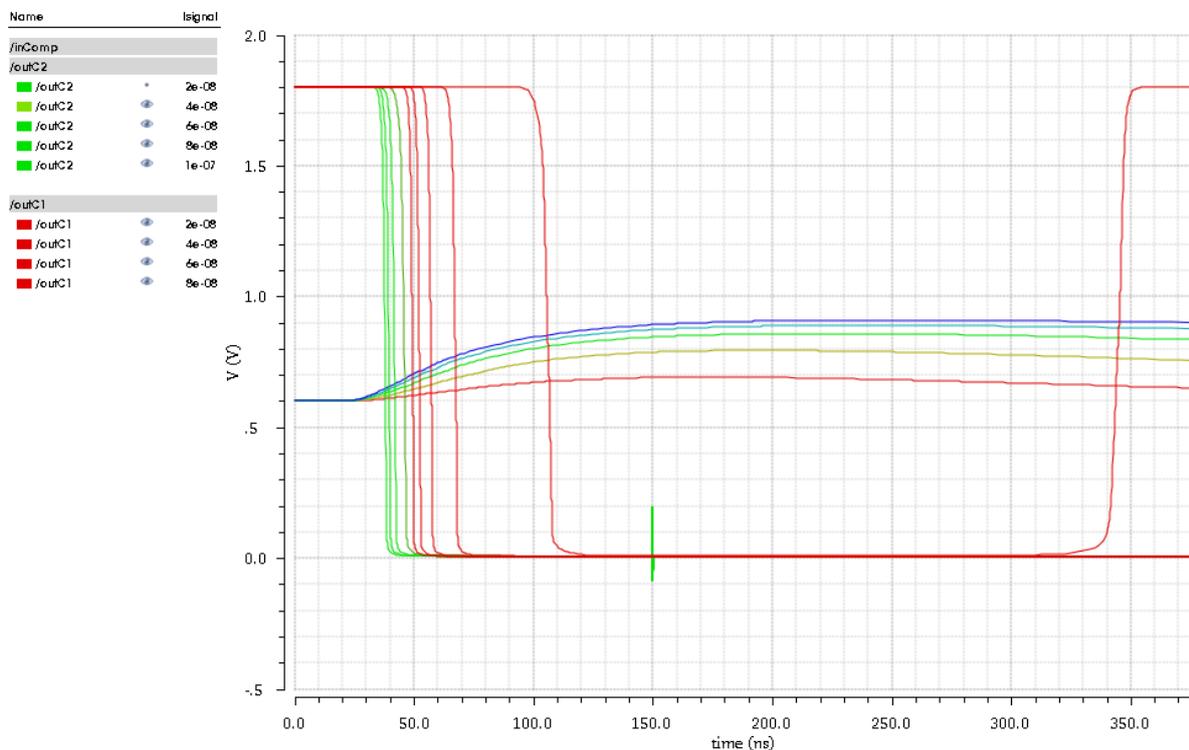
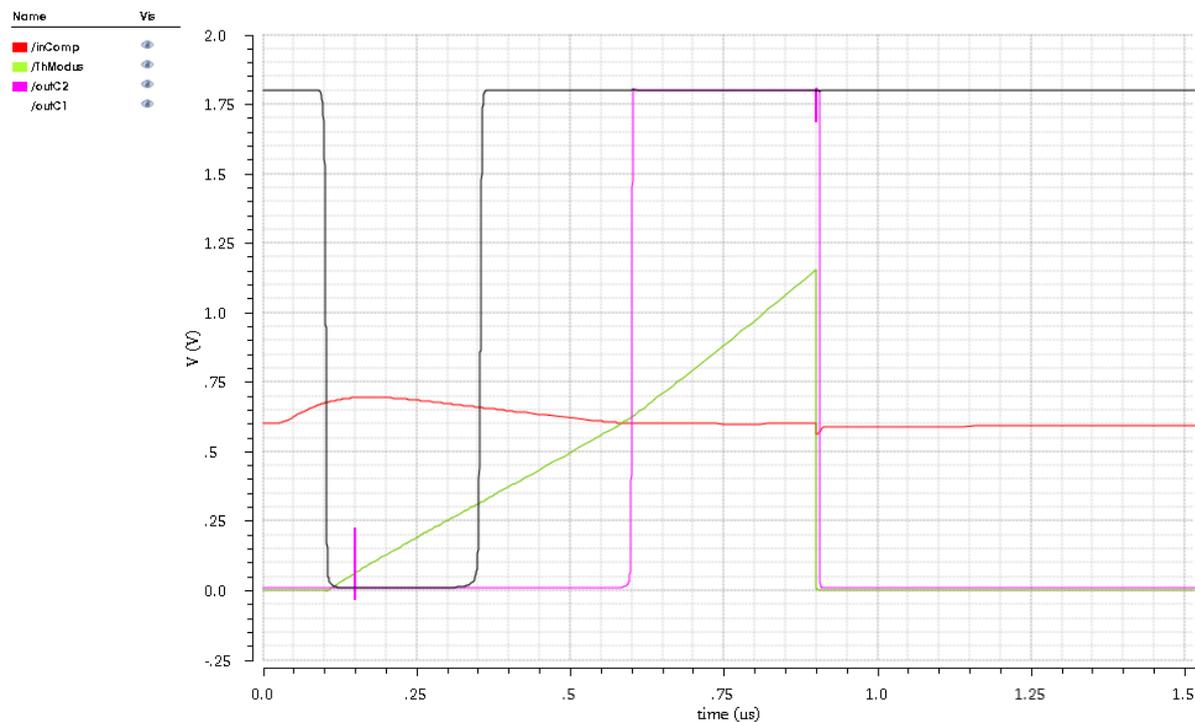


Abbildung 45 Simulation zwei Schwellen bei verschiedenen Eingangsspannungen, Verbesserung des Timewalk Problems

### 3.5.3.2 Simulation Modus II: Amplitudenmessung

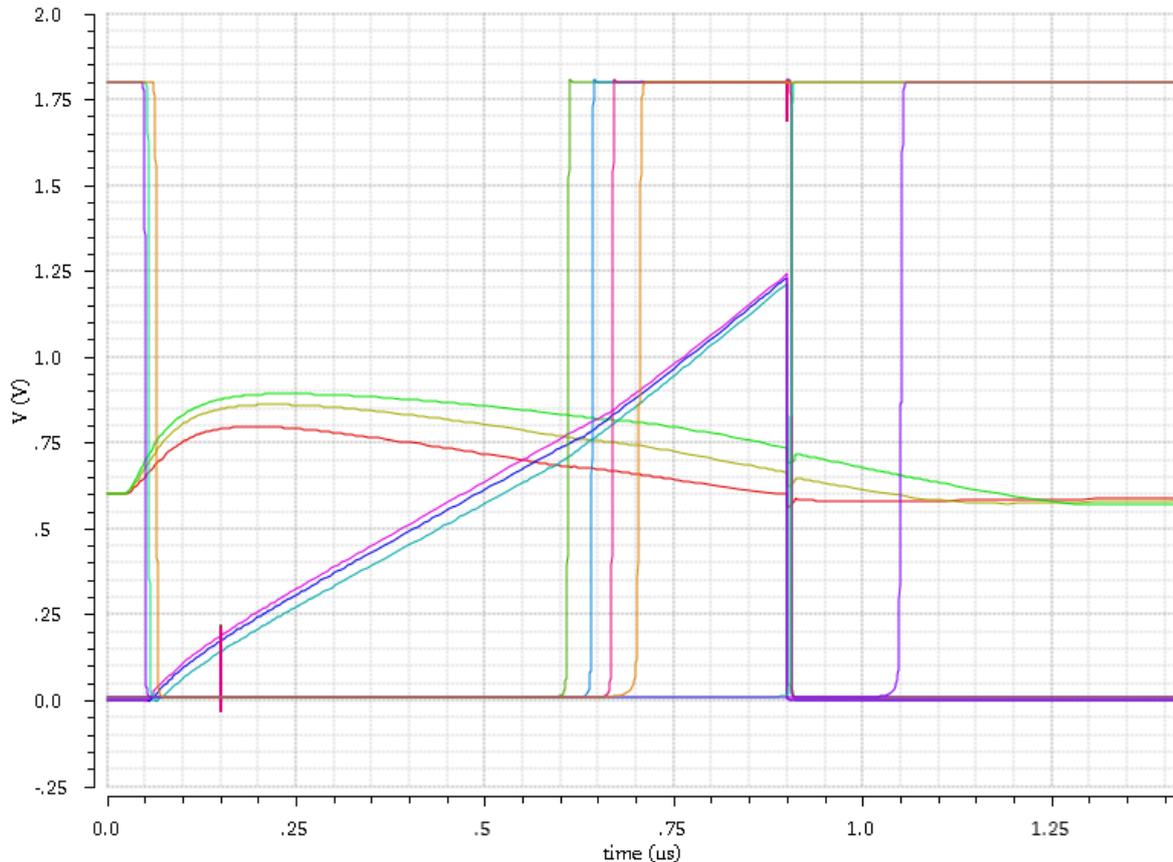
Die Ergebnisse der Anpassungen für die Amplitudenmessung werden ebenfalls durch die Simulation bestätigt. Zunächst die Ergebnisse für einen Eingangsstrom von 20nA in Abbildung 46.



**Abbildung 46 Simulation Komparator mit linear steigender Schwellspannung**

Das rote Signal zeigt den bereits bekannten Verlauf des Eingangssignals. In blau sieht man die Ausgangsspannung des ersten Komparators mit konstanter Schwelle, identisch mit dem Ausgang des Komparators mit höherer Schwelle, zu sehen in pink in Abbildung 44. Die linear steigende Schwellspannung des zweiten Komparators ist in grün dargestellt. Sobald die Schwelle überschritten wurde, beginnt die Schwellspannung zu steigen. Schneidet die steigende Schwellspannung das abfallende Eingangssignal, so schaltet auch der nächste Komparator. Das Ende der steigenden Schwelle wird durch die Logik bestimmt.

Um nun deutlich zu sehen, dass diese Schaltung Rückschlüsse auf die Höhe der Amplitude der Eingangsspannung erlaubt, ist es notwendig eine Simulation mit Sweep zu betrachten. Hierfür wurden die Eingangsströme zu 40nA, 60nA und 80nA gewählt. Die Simulation ist in Abbildung 47 gezeigt und wird unten erläutert.



**Abbildung 47 Simulation Komparator mit linear steigender Schwellspannung, Sweep mit drei verschiedenen Amplituden**

Schön zu sehen sind die verschiedenen hohen Amplituden der Eingangsspannung. Diese fallen unterschiedlich schnell ab. Da der Verlauf der Schwellspannung nahezu identisch ist, schneidet diese die Eingangssignale zu verschiedenen Zeitpunkten. Das blaue, rosa und gelbe Signal sind die zu dem Komparator für Modus II zugehörigen Schaltflanken. Deutlich zu erkennen ist, dass die Schaltzeitpunkte des zweiten Komparators je nach Höhe der Eingangsamplitude verschoben sind. Die niedrigste Amplitude führt als erste zu einem Schalten, je höher die Amplitude, desto weiter verschiebt sich der Schaltzeitpunkt nach rechts. Somit lässt sich eindeutig zuordnen, welche Kennlinie zu welcher Amplitude gehört. Dies kann nun ausgenutzt werden, um ausgehend vom Zeitpunkt des Schaltvorgangs des Komparators Rückschlüsse auf die Amplitude und damit den Eingangsstrom zu ziehen.

### 3.5.3.3 Monte Carlo Simulation Modus I: zwei Schwellen

Bei der Monte Carlo Simulation des Modus I mit den zwei verschiedenen Schwellenspannungen zeigt sich ein ähnliches Bild wie zuvor in der Monte Carlo Simulation eines einzelnen Komparators, siehe 3.4.3.2. Gezeigt wird das Ergebnis der Simulation unten in Abbildung 48. Die erste entscheidende Flanke, wenn das Eingangssignal die Schwelle überschreitet und der Komparator von high auf low schaltet, bleibt vom Mismatch nahezu unverändert. Anders dagegen beim Zurückschalten des Komparators, hier gibt es eine große Verzerrung durch die minimalen Veränderungen der Transistoren. Dieses Verhalten ist jedoch für die Funktionalität der Schaltung ohne Bedeutung, da diese Flanke kein Signal auslöst und daher keinen Beitrag zur Funktion leistet.

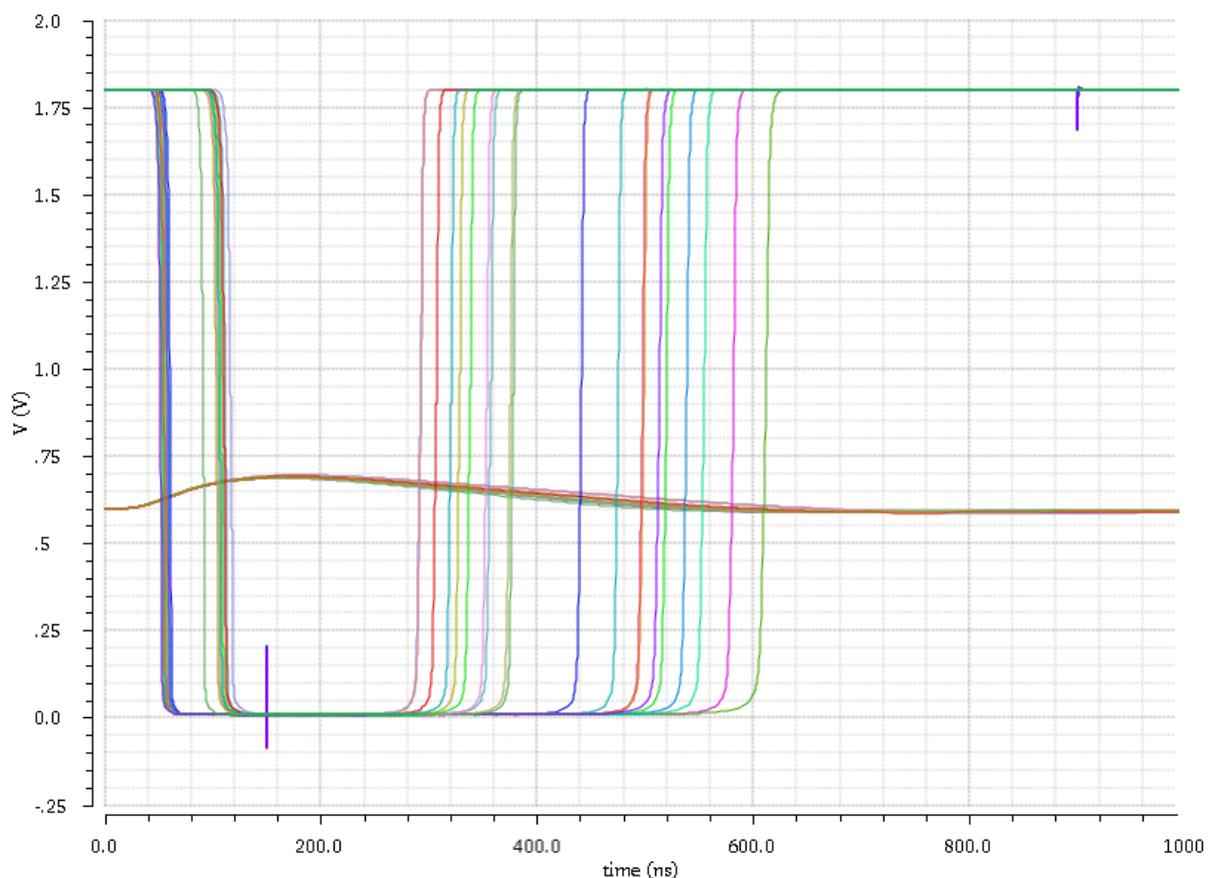


Abbildung 48 Monte Carlo Simulation Modus mit zwei Schwellenspannungen

### 3.5.3.4 Monte Carlo Simulation Modus II: Amplitudenmessung

Auch für den zweiten Modus des ADC mit der linear steigenden Schwellspannung wurde eine Monte Carlo Simulation durchgeführt. Hier zeigt sich sehr schön der Vorteil dieses Entwurfs. Der Schaltvorgang des Komparators mit der steigenden Schwellspannung liegt bei allen Varianten des Monte Carlo Verfahrens sehr eng beieinander. Dies sind in Abbildung 49 die blauen Kurven, sie liegen alle sehr nah beieinander. Dies bedeutet, dass die Flanke des Komparators, mit deren Hilfe später Rückschlüsse auf die Amplitude gezogen werden können, nicht stark durch den Mismatch der Transistoren beeinflusst und damit verfälscht wird. Dieser Modus kann somit zuverlässig eine Einschätzung der Amplitude des Eingangssignals geben.

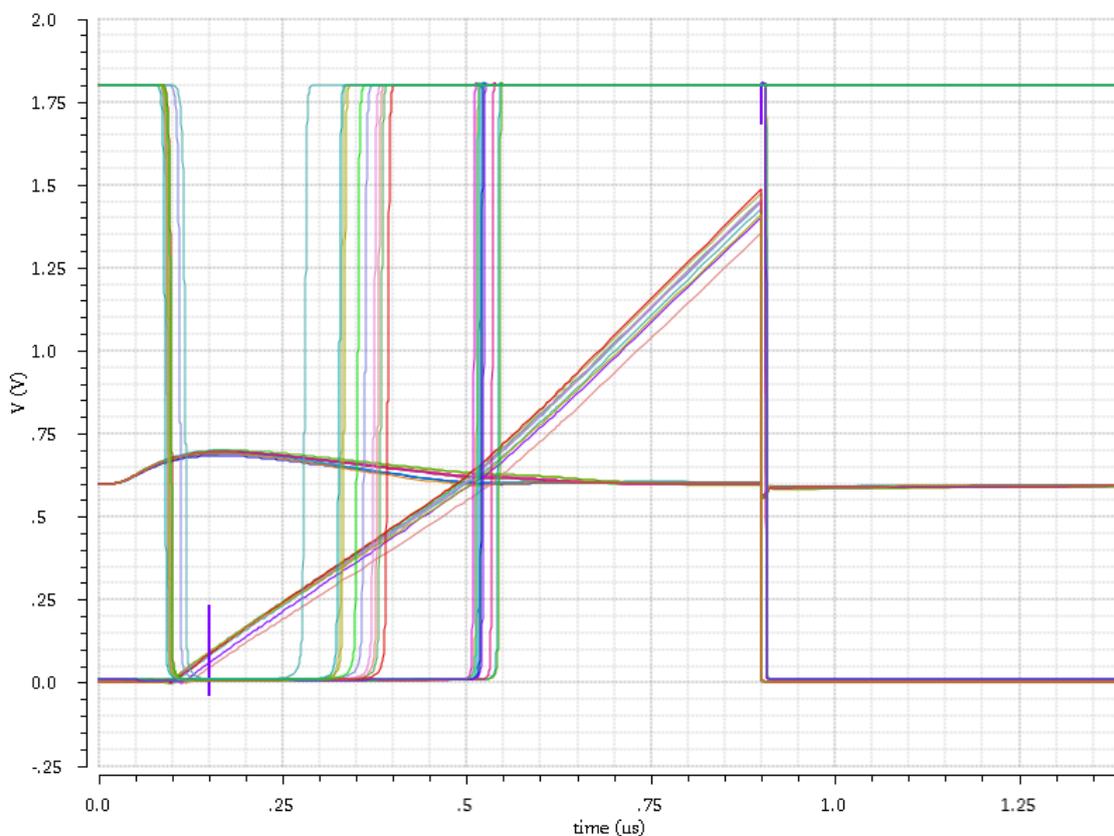
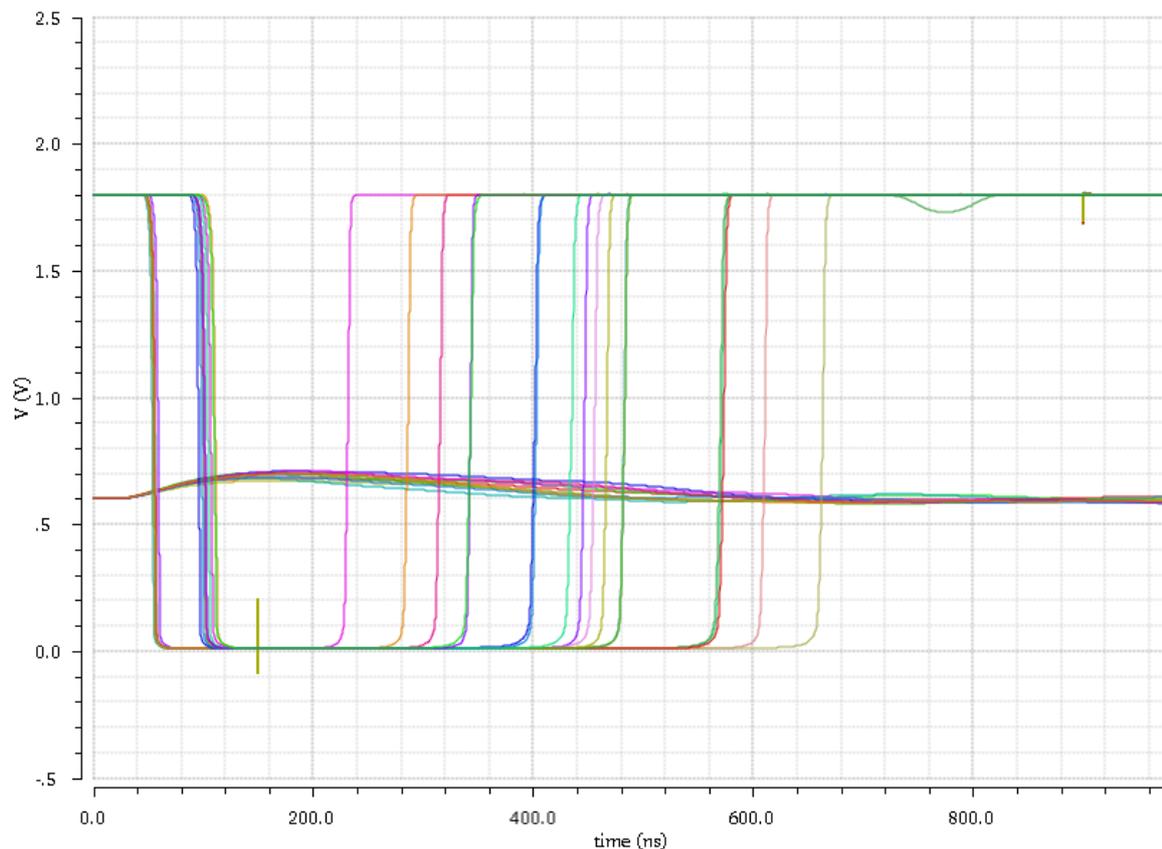


Abbildung 49 Monte Carlo Simulation Modus mit linear steigender Schwellspannung

### 3.5.3.5 Simulation mit Rauschen Modus I: zwei Schwellen

Zusätzlich zu den rauschfreien Simulationen wurden auch Simulationen unter Berücksichtigung des Rauschens gemacht. Diese zeigen, ob die Schaltung auch unter realen, also rauschbehafteten, Bedingungen korrekt funktioniert.

Zunächst die Simulation für den Modus mit den beiden konstanten Schwellspannungen. Auch dieses Ergebnis ist sehr erfreulich. In Abbildung 50 erkennt man die Schaltvorgänge für den Komparator mit der niedrigen Schwelle, der zuerst schaltet, und die Schaltvorgänge des Komparators mit der höheren Schwelle, der kurze Zeit später auf low schaltet. Vergleicht man die Kurven der beiden Komparatoren, so wird deutlich, dass die niedrige Schwelle auch dazu führt, dass es zu einer kleineren zeitlichen Abweichung durch das Rauschen für den Zeitpunkt des Schaltvorgangs kommt. Dass die steigenden Flanken beider Komparatoren stärker durch das Rauschen beeinflusst werden, ist in dieser Anwendung von keiner Bedeutung, da dies die gewünschte Funktionalität nicht beeinflusst.



**Abbildung 50 Rauschsimulation des Modus I mit zwei konstanten Schwellspannungen**

### 3.5.3.6 Simulation mit Rauschen Modus II: Amplitudenmessung

In der folgenden Abbildung 51 wird der Einfluss des Rauschens auf die Simulation des Modus zwei mit steigender, rampenförmiger Schwellspannung gezeigt. Hier ist es von großer Bedeutung, dass die Flanke des Komparators mit der steigenden Schwellspannung nicht verrauscht wird, diese muss verlässlich zum gleichen Zeitpunkt für die gleiche Eingangsamplitude erfolgen. Nur so kann diese für eine zuverlässige Messung verwendet werden. Das Ergebnis der Simulation zeigt, dass der DAC im Modus II genau diese Anforderungen sehr gut erfüllt. Deutlich ist der Schnittpunkt der Schwelle mit dem Eingangssignal erkennbar und hier erfolgt der Schaltvorgang ohne große Verschiebung.

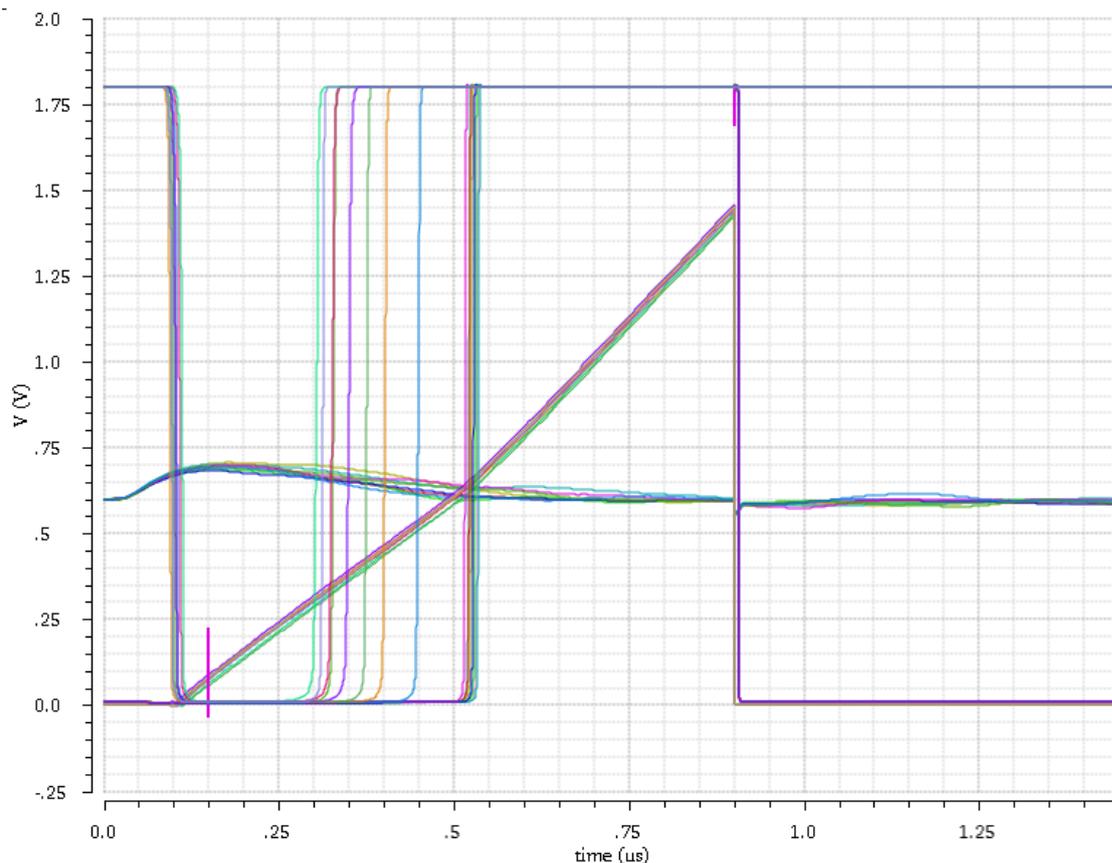


Abbildung 51 Rauschsimulation des Modus II mit Amplitudenmessung

Zusätzlich wird hier nun noch eine Simulation vorgestellt, die zeigt, dass es möglich ist, in diesem Modus bei Rauschen verschiedene Eingangsamplituden eindeutig voneinander zu trennen. In Abbildung 52 ist eine Simulation mit Rauschen und drei unterschiedlichen Eingangsamplituden gezeigt. Die Eingangssignale wurden ausgelöst durch einen Strom von 30nA (rot), 60nA (türkis) und 90nA (grün). Zuerst sieht man die verschiedenen Amplitudenwerte der Eingangsspannungen, je höher der Strom, desto höher die Amplitude. In den gleichen Farben wie die Eingangsspannungen wurden auch die zugehörigen Ausgänge der Schaltung gefärbt. Trotz Rauschens sind diese in drei Bereiche einzuteilen. Als erstes erfolgt bei den roten, zu den 30nA gehörenden, Kennlinien der Schaltvorgang. Anschließend

die türkisfarbene und schließlich in Grün die Kennlinie zu der höchsten Amplitude der Eingangsspannungen. Zwischen den einzelnen Bereichen gibt es jeweils eine kleine Lücke, die trotz Rauschens vorhanden ist. Somit lassen sich aus dem Zeitpunkt des Schaltvorgangs des Komparators mit der steigenden Schwellspannung, diese ist hier in blau dargestellt, Rückschlüsse ziehen auf die Höhe der Amplitude. Zusammenfassend erfüllt dieser Modus voll und ganz die an ihn gestellten Anforderungen.

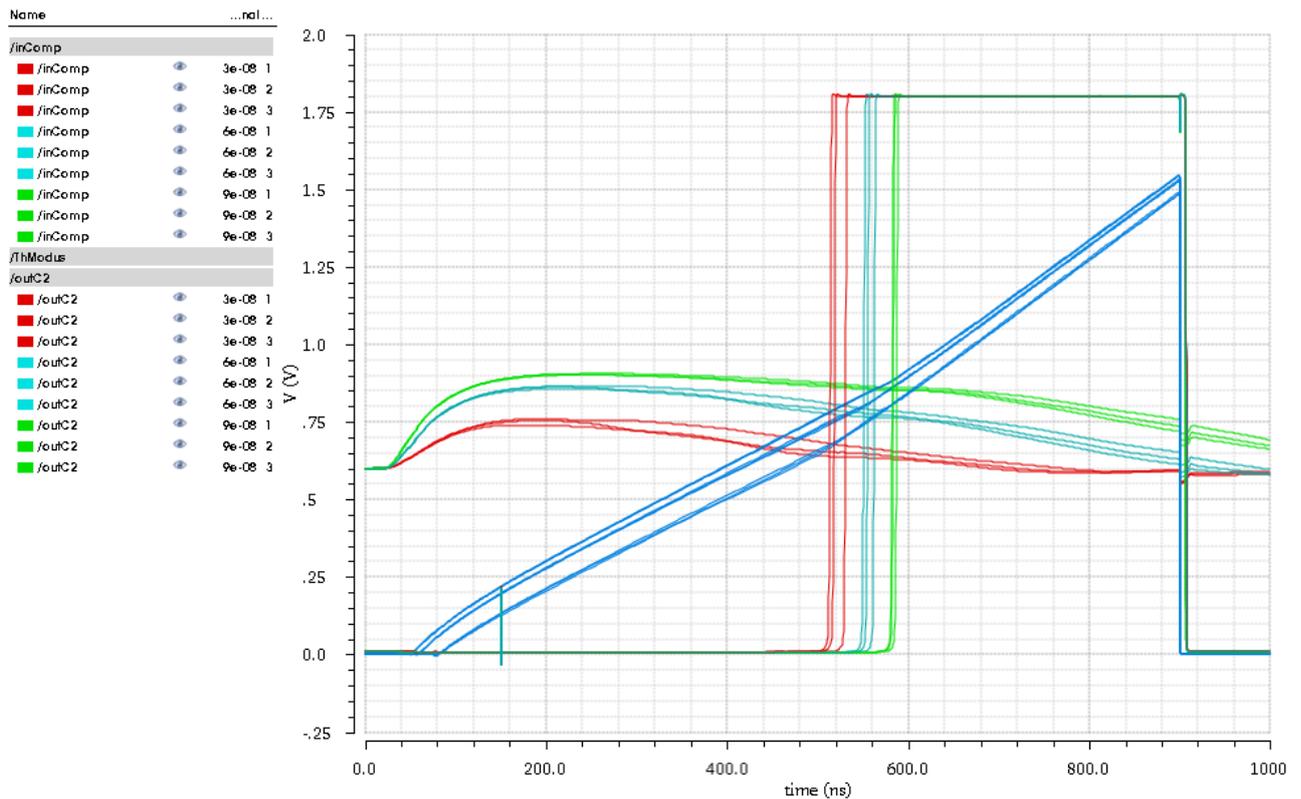


Abbildung 52 Rausch-Simulation im Modus II: Amplitudenmessung mit verschiedenen Eingangssignalamplituden

### 3.6 Element: Scan/Oder-Kette mit fastIn

Die Scanlogik ist ein weiterer Teil, mit dem ich mich ausführlich beschäftigt habe. Die Aufgabe der Scanlogik ist es, den ersten Treffer zu lokalisieren. Kommt es zum exakt gleichen Zeitpunkt zu mehreren Treffern, so wird nur der am weitesten oben im Pixelarray detektierte Treffer ausgelesen. Daher muss eindeutig festgestellt werden, wo sich dieser befindet. Dies wird durch die Scanlogik ermöglicht. In Tabelle 7 sieht man ein Beispiel von hits verschiedener Pixel und dazu erwünschte Ausgangswerte. Eine weitere wichtige Anforderung ist es, dies so schnell wie möglich ermitteln zu können. Daher wurde dieser Teil in besonderer Weise zeitlich optimiert.

<i>Pixel</i>	<i>hit</i>	<i>y</i>	<i>select</i>
1	0	0	0
2	1	1	1
3	0	1	0
4	0	1	0
5	1	1	0
6	0	1	0

Tabelle 7 Beispiel für Funktion der Scanlogik

#### 3.6.1 Konzept

Die gewünschte Funktion lässt sich am einfachsten durch eine oder-Kaskadierung beschreiben. Jeder Eingang ist mit dem nächsten über ein Oder verbunden. Hier ein Beispiel mit fünf Eingängen, die Funktion ist:  $((((f \text{ or } x_1) \text{ or } x_2) \text{ or } x_3) \text{ or } x_4)$ . Das erste Oder dient der Überprüfung des Wertes von  $a$  gegenüber  $f$ . Der Wert von  $f$  müsste hier 0 sein, um eindeutig  $x_1$  ermitteln zu können. Selbstverständlich könnte in der allerersten Stufe auf dieses Oder und  $f$  verzichtet werden und  $x_1$  direkt ausgegeben werden. Das untere Beispiel soll jedoch einen Ausschnitt aus einer sich wiederholenden Struktur zeigen, daher wird das erste Oder hinzugenommen. Auch für die weitere Betrachtung und anschließende Optimierung ist es von Vorteil, ein sich wiederholendes Element zu untersuchen, daher wurde das erste Element auch dort übernommen. Für dieses Beispiel sind 24 Transistoren notwendig.

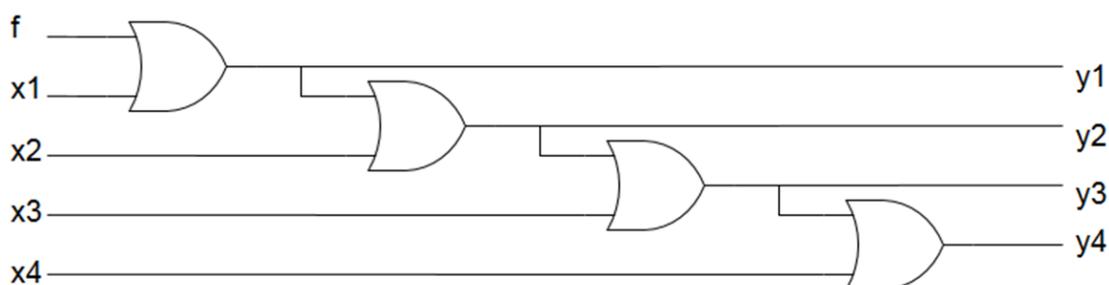


Abbildung 53 OR-Kaskadierung

Nun erfolgt zunächst eine Umformung durch die Boolesche Logik und die Regeln der Schaltalgebra, um eine Realisierung mit weniger Transistoren zu erhalten.

$$\begin{aligned} & \overline{\overline{\left(\left(\left(f \text{ or } x_1\right) \text{ or } x_2\right) \text{ or } x_3\right) \text{ or } x_4}} \\ & \overline{\overline{\left(\left(\left(f \text{ or } x_1\right) \text{ or } x_2\right) \text{ or } x_3\right) \text{ and } \overline{x_4}} \\ & \overline{\overline{\left(\overline{\left(\left(f \text{ or } x_1\right) \text{ or } x_2\right) \text{ and } \overline{x_3}\right) \text{ and } \overline{x_4}} \\ & \overline{\overline{\left(\overline{\left(\overline{\left(f \text{ or } x_1\right) \text{ and } \overline{x_2}\right) \text{ and } \overline{x_3}\right) \text{ and } \overline{x_4}} \\ & \overline{\overline{\left(\overline{\left(\left(f \text{ nor } x_1\right) \text{ and } \overline{x_2}\right) \text{ and } \overline{x_3}\right) \text{ and } \overline{x_4}} \\ & \overline{\overline{\left(\overline{\left(\left(\left(f \text{ nor } x_1\right) \text{ and } \overline{x_2}\right) \text{ and } \overline{x_3}\right) \text{ or } x_4\right) \\ & \overline{\overline{\left(\overline{\left(\overline{\left(\left(f \text{ nor } x_1\right) \text{ and } \overline{x_2}\right) \text{ or } x_3\right) \text{ or } x_4\right) \\ & \overline{\overline{\left(\overline{\left(\left(\left(f \text{ nor } x_1\right) \text{ nand } \overline{x_2}\right) \text{ or } x_3\right) \text{ or } x_4\right) \\ & \overline{\overline{\overline{\left(\overline{\left(\left(\left(f \text{ nor } x_1\right) \text{ nand } \overline{x_2}\right) \text{ or } x_3\right) \text{ or } x_4\right) \\ & \overline{\overline{\overline{\left(\overline{\left(\overline{\left(\overline{\left(f \text{ nor } x_1\right) \text{ nand } \overline{x_2}\right) \text{ or } x_3\right) \text{ and } \overline{x_4}} \\ & \overline{\overline{\overline{\left(\overline{\left(\left(\left(f \text{ nor } x_1\right) \text{ nand } \overline{x_2}\right) \text{ nor } x_3\right) \text{ and } \overline{x_4}} \\ & \overline{\overline{\overline{\left(\overline{\left(\left(\left(f \text{ nor } x_1\right) \text{ nand } \overline{x_2}\right) \text{ nor } x_3\right) \text{ nand } \overline{x_4}} \end{aligned}$$

Dies bringt einen Vorteil, da die Ausgänge der Pixel, gemeint die hits, bereits in nicht-negierter und negierter Form vorliegen. Dies führt zu einer notwendigen Gesamtanzahl von 20. Abbildung 54 zeigt die zu der Gleichung passende Logik.

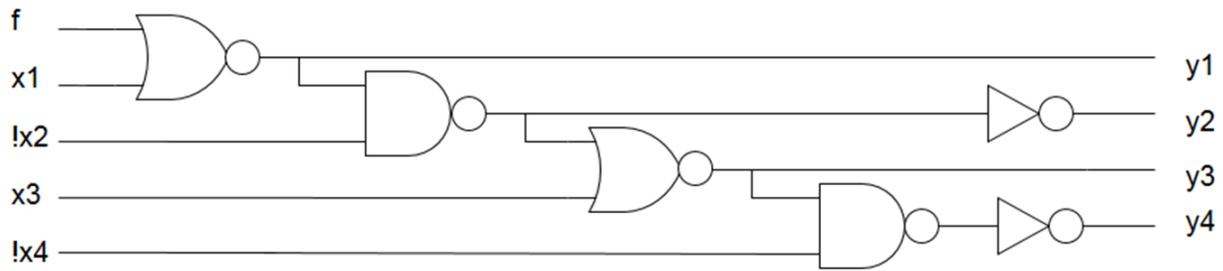


Abbildung 54 NOR/NAND-Kaskadierung

Man erkennt, dass sich die Struktur immer für zwei Eingänge wiederholt, d.h. später werden sich zwei Pixel eines dieser Scanelemente teilen. Dies spiegelt sich auch in dem in 5.5 gezeigten Layout wider. Nach dieser ersten Optimierung habe ich eine weitere Optimierung durchgeführt. Ziel ist es, die Laufzeit deutlich zu verkürzen. Hierfür wird ein weiteres Zwischensignal benötigt, im Folgenden fastIn genannt. Grundidee des fastIn ist es, es zu ermöglichen, dass man nicht mehr jede Stufe durchlaufen muss, um den Ausgang der Scanlogik zu erhalten. Vielmehr möchte man ein schnelles und möglichst zeitgleiches Schalten der Ausgänge der einzelnen Pixellogikelemente. Das hier entwickelte Konzept sieht eine Gruppierung der Elemente vor. Im Fall des Mu3e ergibt sich pro Spalte eine optimale Gruppengröße von 20, bei 10 Gruppen pro Spalte. Jede Gruppe teilt sich ein fastIn. Dieses wird bei dem jeweils ersten Element einer Gruppe auch als Signal für den Eingang f in Abbildung 54 verwendet. Das fastIn wird dann mit jedem Ausgang durch ein Oder Element verbunden. Dies wird realisiert durch Inverter, Nor und Nutzung der Tatsache, dass ein Teil der, aus der zuvor vorgestellten Logik kommenden, Ausgänge bereits negiert ist. Dies hat nun einen großen Vorteil: Diese Oder Elemente können für alle Ausgänge gleichzeitig überprüft werden. Ist nun das fastIn bereits 1, dann werden alle Ausgänge der Gruppe, die dieses fastIn nutzt, ebenfalls 1. Damit hat man sich den langsamen Durchlauf durch die Kaskadierung der in Abbildung 54 gezeigten Elemente gespart. Die gerade beschriebene Verschaltung ist in Abbildung 55 gezeigt. Auf den Teil zwischen den y- Ausgängen und den select Ausgängen wird später eingegangen.

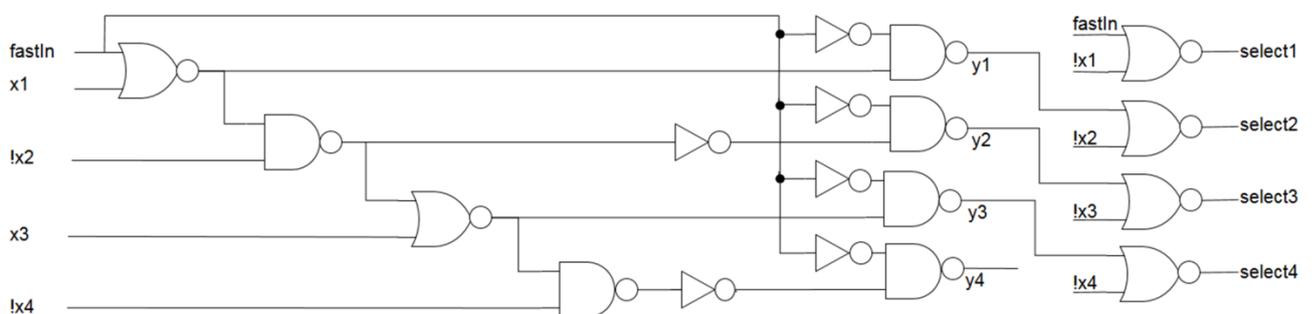


Abbildung 55 Scanlogik mit vier Eingängen

Bei der allerersten Gruppe einer Spalte muss fastIn immer null sein. Nun zu der Frage, wie das fastIn bei den kommenden Gruppen gewählt wird. Das fastIn muss eins sein, falls es in einer der vorherigen Gruppen bereits einen hit gab, sonst muss es immer null sein. Diese Voraussetzung wird genau von dem y des letzten Elements der vorherigen Gruppe erfüllt. Daher wird als neues fastIn der nächsten Gruppe immer das letzte y der vorherigen Gruppe gewählt. Somit ist nun für jede Gruppe klar, wie das fastIn korrekt gewählt werden muss.

Nun zu dem select Ausgang. Dieser soll für den ersten hit eine eins liefern und für alle folgenden Elemente, unabhängig davon, ob diese null oder eins sind, null ergeben. Dies wird hier umgesetzt durch ein Nor. In dieses wird das fastIn und die negierte Form des Eingangs, also !x, gegeben. Das x muss nicht hier negiert werden, da man aus dem Logikteil, der der Scanlogik voran geschaltet ist, sowohl die Signale x als auch die negierten !x erhält. Dieser Teil der Scanlogik profitiert ebenfalls von den Vorteilen des fastIn, da der y Wert verwendet wird und dessen Laufzeit durch das fastIn stark verkürzt wurde.

Die Zeitersparnis, die durch das fastIn bewirkt wird, ist abhängig von der Gruppe, in der der hit erfolgt. Je weiter oben in einer Spalte sich dieser befindet, desto größer ist die Zeitersparnis, da dann mehr Gruppen von dem fastIn profitieren können.

### 3.6.2 Schaltplan

Die logischen Elemente, die für die Realisierung der Scanfunktion nötig sind, wurden auf Layoutebene entworfen und so für ein späteres Zusammenfügen der Gatter speziell für die Maße des Mu3e Chips optimiert. Dies sieht man in 5.5, in dem das Layout der Scanlogik genauer vorgestellt wird. In Abbildung 66 sieht man den bereits beschriebenen Schaltplan, er wurde hier realisiert für zwei Gruppen mit je acht Elementen. Gut erkennbar ist die Verschaltung des letzten y-Wertes als fastIn für die nächste Gruppe.

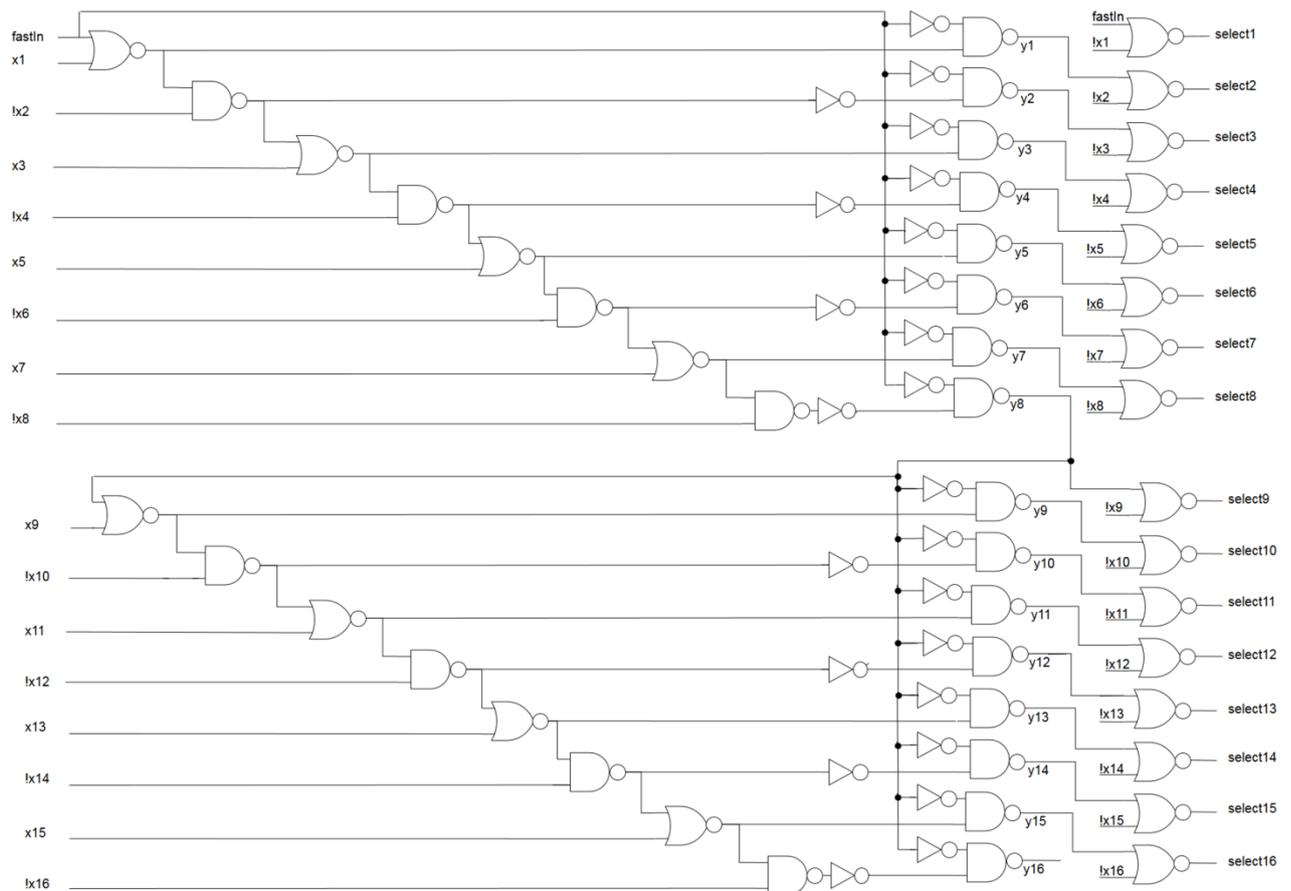


Abbildung 56 Scanlogik, zwei Gruppen mit je acht Eingängen

### 3.6.3 Simulation

#### 3.6.3.1 Ideale Simulation

Die Simulationsergebnisse zeigen im Folgenden die Korrektheit der Überlegungen bezüglich des fastIn Signals zur Verkürzung der Laufzeit der Scanlogik.

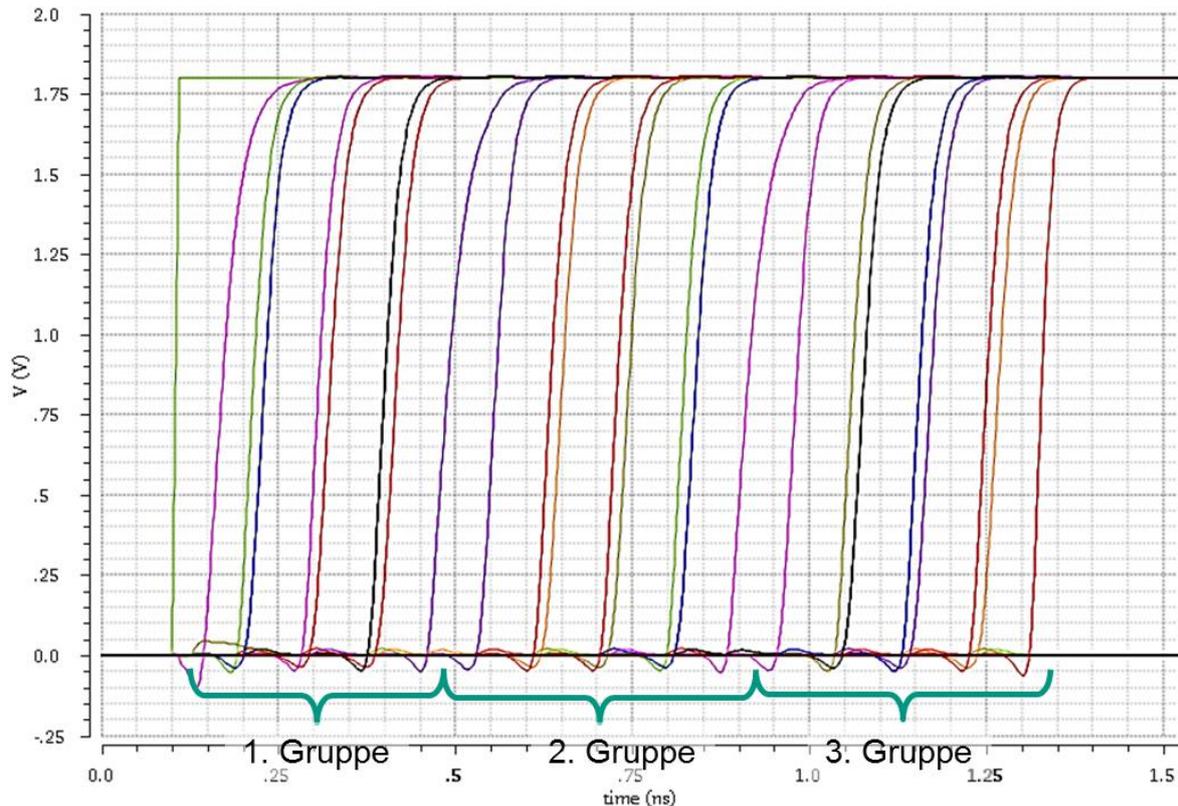


Abbildung 57 Simulation Scanlogik, ohne fastIn, drei Gruppen mit je acht Elementen

In Abbildung 57 ist das Ergebnis einer Simulation von drei Gruppen mit je 8 Elementen gezeigt. Hier wurde das fastIn nicht verwendet. Es zeigt also den Verlauf der Ausgangssignale durch die Kaskadierung ohne die durch diese Arbeit entworfene Optimierung. Der erste Treffer wurde von dem zweiten Element der ersten Gruppe detektiert. Die Zuordnung der Signale der Simulation von links nach rechts ist wie folgt: x2, select2, y2, y3, y4 etc. Deutlich ist der Nachteil der Kaskadierung zu sehen: nacheinander schaltet jeder y Ausgang auf eins und bis das letzte y eins ist, müssen alle durchlaufen werden. Die leichte Gruppierung der Signale der Simulation in zweier Gruppen stammt daher, dass sich die Struktur, wie 3.6.1 erklärt, in Zweierschritten wiederholt. Daher wiederholen sich die zeitlichen Abstände der Ausgangssignale auch hier in Zweierschritten.

Nun der exakt gleiche Simulationsaufbau, diesmal jedoch mit der Optimierung durch das fastIn. Das Ergebnis zeigt Abbildung 58.

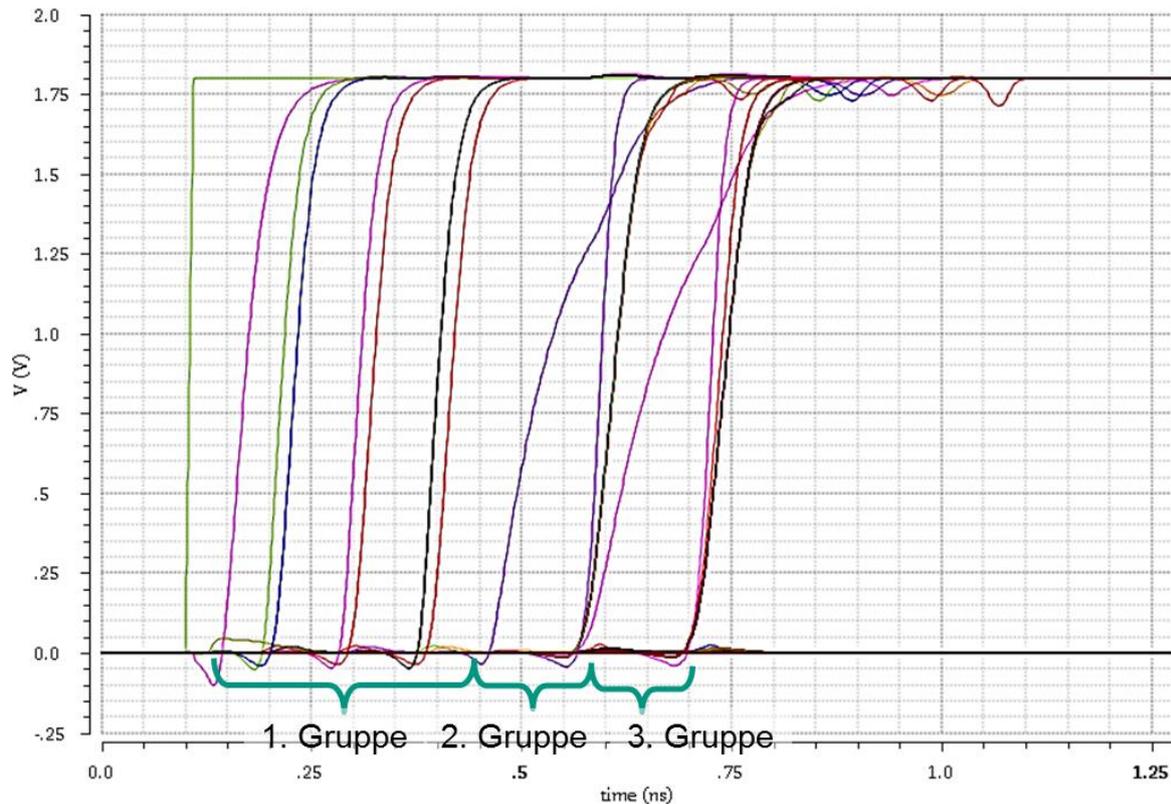
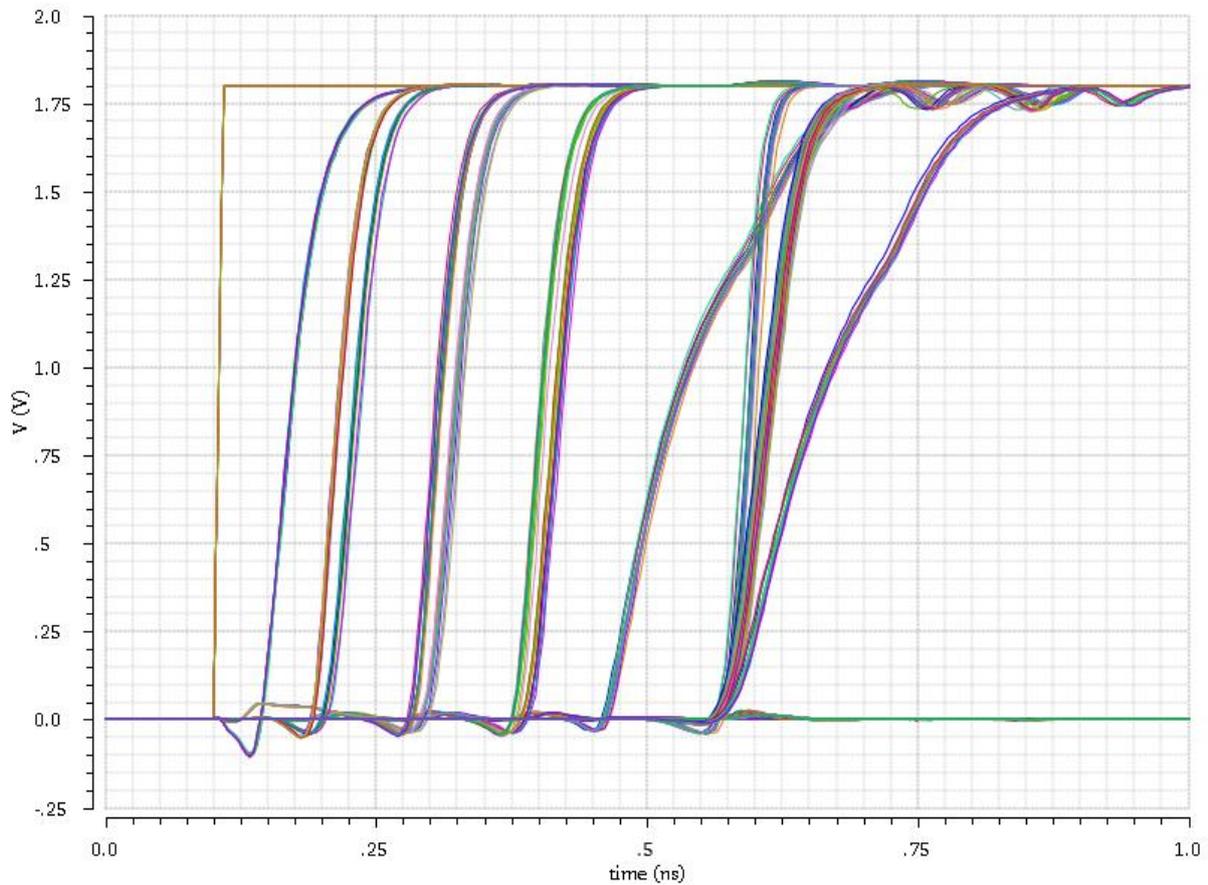


Abbildung 58 Simulation Scanlogik, mit fastIn, drei Gruppen mit je 8 Eingängen

Auch hier erfolgt der Treffer bei  $x_2$ , dem zweiten Pixel der ersten Gruppe. Hier sieht man nun sehr deutlich, dass nachdem die erste Gruppe normal durchlaufen wurde, es zu einer Verkürzung der folgenden Laufzeiten kommt. Bei der zweiten und dritten Gruppe liegen die Ergebnisse praktisch zeitgleich am Ausgang an. Das schräg verlaufende Signal zwischen den Gruppen ist das letzte  $y$  der vorherigen Gruppe, das in der nächsten Gruppe als fastIn verwendet wird. Deutlich ist im Vergleich zu Abbildung 57 die verkürzte Laufzeit. Bereits für nur drei Gruppen ergibt sich eine Einsparung von 40%. Später werden 10 Gruppen je Spalte verwendet, daher wird hier die Zeitersparnis bei einem Hit in den oberen Gruppen noch deutlich größer ausfallen. Insgesamt hat sich durch viele Simulationen das erarbeitete Konzept zur Optimierung bestätigt.

### 3.6.3.2 Monte Carlo Simulation

Die Scanlogik wurde auf Mismatch Probleme durch eine Monte Carlo Simulation getestet. Hier wird eine Reihe von Simulationen mit jeweils zufälligen kleinen Abweichungen der Maße der Transistoren durchgeführt. Die Schar der Ergebnisse zeigt Abbildung 59. Es kommt zu keiner Beeinträchtigung der Scanlogik durch den Mismatch.



**Abbildung 59 Monte Carlo Simulation der Scanlogik**

## 4 Digital Design

### 4.1 Workflow

Während dieser Arbeit habe ich mich auch intensiv mit dem Workflow des digitalen Designs beschäftigt. Dieses unterscheidet sich grundlegend von dem bisher gezeigten Vorgehen, bei dem Schematic und Layout manuell erstellt wurden. In diesem Workflow geschieht dies automatisch. Der ganze Ablauf basiert auf einer Hardware Description Language, hier wurde ausschließlich verilog verwendet. Eine andere weit verbreitete mögliche Hardware Description Language wäre VHDL. Eine Hardware Description Language (HDL) ermöglicht eine abstrakte Beschreibung der Funktion der digitalen Schaltung und generiert davon ausgehend einen Schaltplan und das Layout. Dieses Verfahren wurde für den Logikteil am Ende einer Spalte und den Logikteil der Peripherie verwendet. Anhand Abbildung 60 wird der genaue Ablauf dieses Workflows gezeigt.

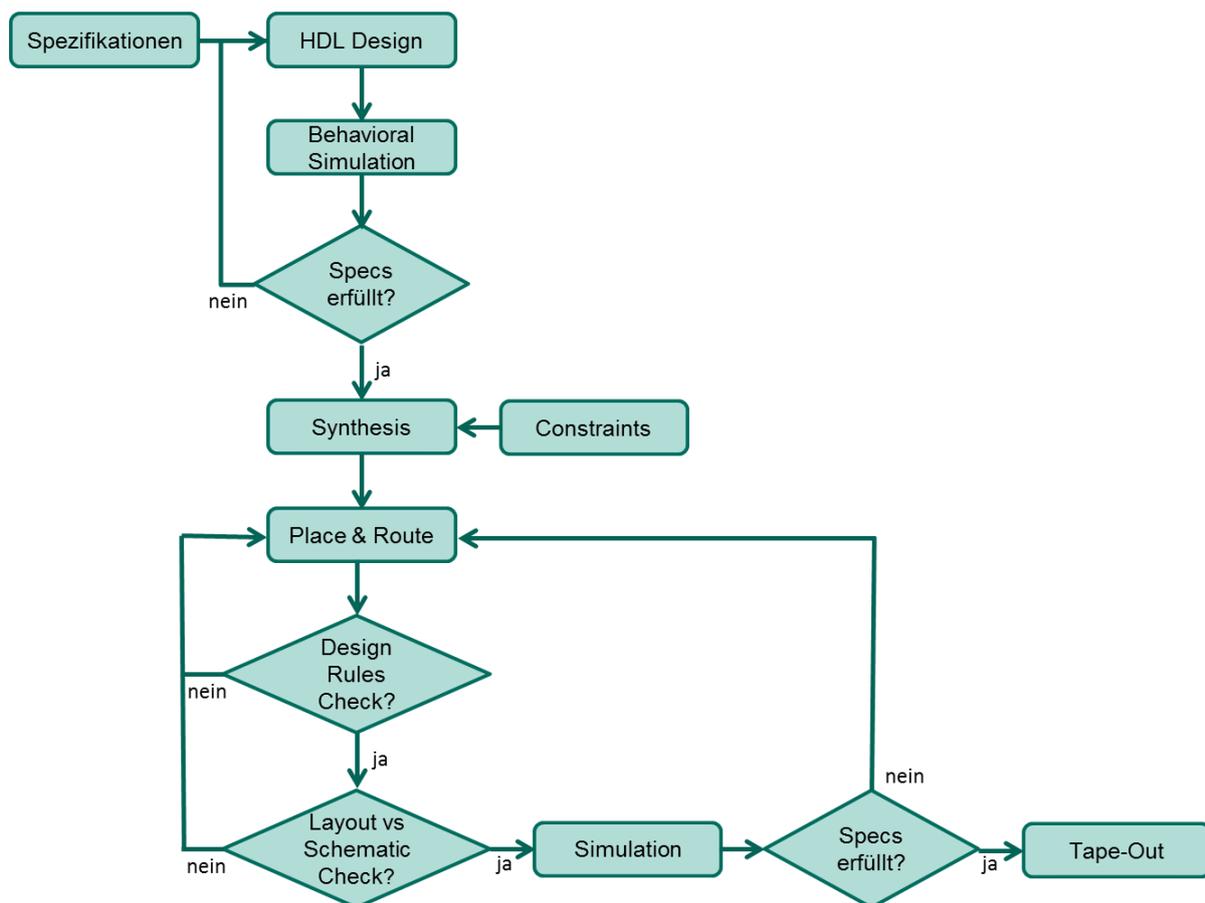


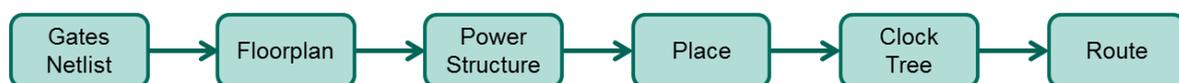
Abbildung 60 Workflow digital Design

Zunächst ermittelt man die Spezifikationen. Anschließend erfolgt das Design mit einer HDL. Hierfür werden Skripte erstellt, die das genaue Verhalten der späteren Schaltung repräsentieren. Dies ist möglich, da HDL anders als zum Beispiel

Softwareprogrammiersprachen eine Gleichzeitigkeit erlaubt. Mithilfe dieses Codes kann eine Verhaltenssimulation durchgeführt werden. Mit deren Ergebnis wird geprüft, ob die Spezifikationen erfüllt werden. Ist dies nicht der Fall, wird das HDL Design, also der Code, überarbeitet und erneut simuliert. Verliefe die Simulation erfolgreich, so kommt der nächste Schritt: die Synthesis. Hier werden mit Berücksichtigung der Constraints die Gates und Netzlisten bestimmt.

Anschließend erfolgt der nächste große Schritt des digitalen Design: Place & Route. Da sich dieser Schritt in mehrere einzelne Prozesse untergliedert, werden diese im Anschluss genauer vorgestellt. Im Place & Route Schritt wird das Layout erzeugt. Dieses muss nun den gleichen Tests wie bei dem zuvor gezeigten Workflow unterzogen werden. Als erstes erfolgt der Design Rules Check und dann bei erfolgreichem Durchlauf, der Layout vs. Schematic Check. Hier wird geprüft, ob das erzeugte Layout dem durch die Synthesis erstellten Schematic entspricht. Im nächsten Schritt erfolgt eine erneute Simulation der Funktion der Schaltung. Die Ergebnisse müssen belegen, dass die Anforderungen des ersten Schrittes erfüllt werden, sonst kommt es zu einem weiteren Iterationsschritt. Sind alle Stufen bewältigt, so erhält man auch hier das Tape-out, welches dem Hersteller zur Fertigung des Chips übergeben wird.

Nun zu dem genaueren Ablauf des Place & Route Vorgangs, siehe Abbildung 61. Mithilfe der Netzlisten wird der Floorplan erstellt. Dieser legt die Größe fest, die das Layout einnehmen soll. Danach wird die Struktur für die Power Supply angelegt. Hier werden sowohl die Versorgungsspannung als auch die Masseleitungen gelegt. Diese werden gleichmäßig auf den Floorplan verteilt. Nun erfolgt die Platzierung der für die Schaltung notwendigen Gatter. Die Optimierung dieser Platzierung verläuft in mehreren Iterationsschritten und wird ebenfalls unter den Überbegriff des Place-Vorgangs gefasst. Daran im Anschluss wird der Clock Tree gelegt. Nun im letzten Schritt erfolgt das Routing aller Elemente, also die Verbindung der zuvor platzierten Elemente untereinander und aller Verbindungen zur Power Structure und zum Clock Tree. Nach diesem Schritt ist der Vorgang des Place & Route abgeschlossen und es liegt nun ein fertiges Layout vor. Dieses muss nun noch die DRC (Design Rules Check) und LVS (Layout versus Schematic Check) erfüllen. Das Vorgehen nach diesem Schritt wurde bereits oben erklärt.



**Abbildung 61 Ablauf von Place & Route**

## 4.2 Aufbau des Digitalteils

### 4.2.1 Überblick

Der digitale Teil in der Chipperipherie besteht aus mehreren Elementen. Hier wird nun ein Überblick über den Aufbau gegeben und anschließend die State Machine als zentrales Element detailliert vorgestellt.

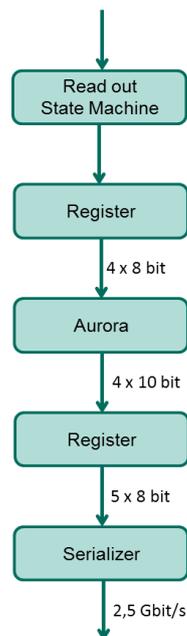


Abbildung 62 Überblick Digitalteil der Chipperipherie

### 4.2.2 State Machine

Der digitale Teil in der Peripherie beinhaltet als zentrales Element eine State Machine (auch Zustandsautomat genannt). Im Folgenden werden die einzelnen Zustände und Abläufe vorgestellt. Anschließend folgt ein Beispiel, das den Ablauf der Zustände und Ereignisse der State Machine veranschaulicht.

Die Aufgabe der hier verwendeten State Machine im digitalen Teil, zu sehen in Abbildung 63, ist es, die Daten eines Pixel, bei dem es einen hit gegeben hat, auszulesen. Beginnend mit dem default Zustand PD1 wird der Ablauf bis zum Auslesen des Wertes des Pixelelementes dargestellt. Im Zustand PD1 wird ein Pulldown Signal aktiviert, das die Adressenleitungen auf Masse zieht. Nach dem Zustand PD1 folgt PD2. Die Zustände mit dem Index 2 repräsentieren Pausen oder Trennzustände. Anschließend wird von hier weiter zu LoadColumn1 gewechselt. In diesem Zustand wird die Spalte ausgelesen, gefolgt von einer Pause, LoadColumn2. Anschließend folgt der erste Zustand, von dem aus es mehrere Möglichkeiten an folgenden Zuständen gibt: Load Pixel1. Falls es keine hits im letzten Frame gegeben hat und gleichzeitig der Counter slowdown nicht seinen maximalen Wert

slowdownend erreicht, verbleibt man in diesem Zustand und erhöht lediglich den Counter slowdown um eins. Ist jedoch ein hit in dem letzten Frame vorhanden, so erfolgt ein Übergang in den nächsten Zustand, der Ablauf erfolgt „schneller“ und wird nicht durch warten in einem Zustand verzögert. Ebenfalls einen Wechsel in diesen Zustand gibt es, falls der Counter slowdown seinen Maximalwert erreicht hat. Nun werden die Pixel „geladen“ – d.h. die hits werden vom ersten ins zweite hit-flag kopiert. Falls es bis jetzt zu keinem hit gekommen ist, so wechselt man in den default Zustand PD1. Gab es jedoch einen oder mehrere hits, so werden die hits in Read Column1 von der Spalte in die end-of-column Logik kopiert. Anschließend wird in Read Column2 (Pause) gewechselt. Falls es hits in end-of-column gibt und der Counter macyc noch nicht seinen Maximalwert maxcycend erreicht hat, wird zurück in Read Column1 gegangen. Falls es keine hits gibt oder maxcycend erreicht wurde, erfolgt ein Wechsel in PD1 und der Counter wird zurückgesetzt. Bei jedem Wechsel zurück in PD1 werden beide verwendeten Counter wieder auf null gesetzt, um den nächsten Durchlauf vorzubereiten.

Nun zu einem beispielhaften Ablauf der State Machine. In Tabelle 8 ist in der ersten Zeile zu sehen, ob gerade Daten (Dat) gesendet werden oder nicht, das Codewort für „keine Daten“ ist: „FF“. In der zweiten Zeile ist der Zustand aufgeführt, in dem sich die State Machine gerade befindet. Die dritte Zeile zeigt, ob in diesem Moment ein hit gefunden wurde oder nicht. Zu Beginn befindet sich die State Machine im Default Zustand PD. Anschließend wechselt sie in Load Column (LC), in diesem Takt gibt es einen hit, genau wie auch im Folgenden. Nachdem die Spalte geladen wurde, erfolgt der Wechsel in Load Pixel (LP). Da es einen hit gibt, erfolgt der direkte Wechsel in Read Column (RC) ohne Wartezeit in Load Pixel. In Read Column werden nun die Daten weitergeben, zu sehen in der ersten Zeile. Da kein weiterer hit in diesem Moment gefunden wurde, kehrt die State Machine in den Zustand PD zurück. Nun erfolgt erneut die Abfolge von Load Column und Load Pixel, da es diesmal jedoch keinen hit gibt, wartet man im Zustand Load Pixel bis der Counter für die Wartezeit (slowdown) seinen Maximalwert erreicht. Da nun immer noch kein hit vorliegt, wird zurück in PD gewechselt. Nun im nächsten Fall werden erneut Load Column und Load Pixel durchlaufen und, da ein hit vorliegt, gleich weiter in Read Column gewechselt und die Daten weiter gegeben. Da es immer noch einen hit gibt, bleibt die State Machine im Zustand Read Column bis der Maximalwert des Counters maxcyc, der steuert, wie lange man in Read Column verbleiben darf, seinen Maximalwert erreicht. Dann erfolgt der Übergang zurück in PD.

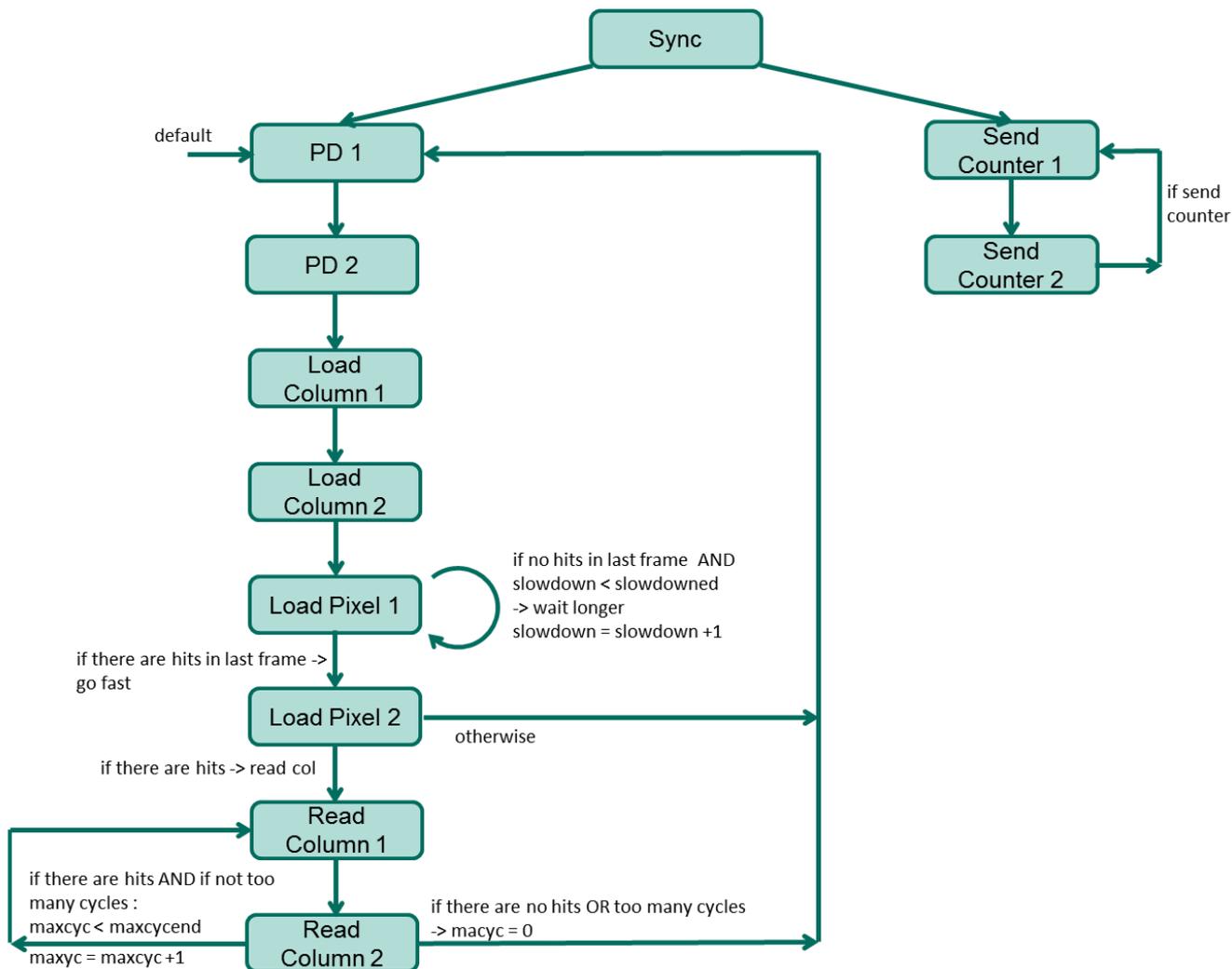


Abbildung 63 State Machine

FF	FF	FF	Dat	FF	FF	FF	FF	FF	FF	FF	Dat	Dat	Dat	FF	FF	Dat
PD	LC	LP	RC	PD	LC	LP	LP	PD	LC	LP	RC	RC	RC	PD	LC	LP
	hit	hit							hit							

Tabelle 8 Beispiel für Ablauf der State Machine

## 5 Layout

### 5.1 Anforderungen

Für alle in dieser Arbeit vorgestellten analogen und digitalen Elemente wurde ebenfalls im Zuge der Arbeit das Layout entworfen.

Die an das Layout gestellten Anforderungen werden hier vorgestellt und gleichzeitig ein Einblick gegeben, wie ein gutes Layout erstellt werden sollte. Kapitel 2.5 erklärt die Grundlagen für den Entwurf eines Layouts. In 5.2 wird ein Ansatz für eine optimale Durchführung des Layout Vorgangs vorgestellt.

Wichtig ist es hier, eine möglichst geschickte Anordnung zu finden, um möglichst platzsparend und mit möglichst wenigen und möglichst kurzen Verbindungen die einzelnen Transistoren bzw. Grundgatter zu verbinden. Durch die Anordnung lässt sich vermeiden, dass man für eine Ebene an Elementen mehrere Metalllagen benötigt. Im besten Fall wird für eine „Elementebene“ jeweils nur eine Lage verwendet. Mit „Elementebene“ sind hier die einzelnen Teile gemeint, in welche das gesamte Layout unterteilt wurde. Bei dem Design der Scanlogik sollte am besten nur eine Lage verwendet werden, genauso für den Komparator.

Erst wenn man nun mehrere Elemente zusammengefügt hat und diese verbindet (sozusagen eine Abstraktionsebene höher), sollte auch eine höhere Metalllage verwendet werden. Selbstverständlich sollte niemals eine höhere Ebene verwendet werden, falls es möglich ist, die gewünschte Verbindung mit einer tieferen Lage zu realisieren. Auf dieser Ebene des Zusammenfügens muss jedoch häufig auf die nächst höhere Lage zurückgegriffen werden, diese sollte daher in den unteren Ebenen (kleinere Elemente) noch nicht verwendet worden sein, um diese Wegfreiheiten nicht einzuschränken. Auch sollte man immer abwechselnd eine Metalllage vor allem für horizontale und die nächste Lage primär für vertikale Verbindungen nutzen.

Dies alles führt dazu, dass man im nächsten Schritt, wenn die nächsten Teile integriert werden und man sozusagen eine Ebene aufsteigt, mehr Freiraum für die Verbindungen und deren optimale Position hat. Wenn man sich bereits zu Beginn streng an die Vorgaben hält und einen Überblick über die Gesamtheit der Elemente und der benötigten Verbindungen hat, so vermeidet man spätere Komplikationen, die unter Umständen zu großen „Umbauten“ führen. Natürlich müssen auch alle Designregeln des Prozesses eingehalten werden, damit das Layout später auch produzierbar ist.

Insgesamt kann man folgende Eigenschaften eines „guten Layouts“ festhalten: möglichst platzsparend, den Platz optimal nutzen und die Verwendung möglichst niedriger Metalllagen, so dass es später genügend Möglichkeiten gibt, die Signale mithilfe der höheren Lagen nach außen zur Peripherie zu übertragen.

## 5.2 Ansatz für Optimierung

Ein entscheidender Schritt im Gestalten eines guten Layouts ist die Platzierung der einzelnen Transistoren bei einem analogen Schaltungsteil und der logischen Gatter bei einem digitalen Teil. Mit dieser Platzierung wird bereits eine weitreichende Designentscheidung getroffen, sie legt fest, wo Inputs und Outputs liegen, und vor allem, wie groß die Verbindungen zwischen den Transistoren/Gattern sein werden.

### 5.2.1 Idee

Das Problem der Platzierung gleicht einer Aufgabe aus der Graphentheorie. Für die Platzierung der Transistoren und Gatter wurde in dieser Arbeit eine Methodik angewendet, die an den Kernighan-Lin Algorithmus des Partitionierungsproblems angelehnt ist.

Der Kernighan-Lin Algorithmus wird hier vorgestellt, allerdings nur Teile, die für die spätere Optimierung der Anordnung verwendet wurden. Für eine genaue Betrachtung sei auf [10] verwiesen. Der Kernighan-Lin Algorithmus ist ein Vorgehen zur Lösung des Partitionierungsproblems. Es wird in der Regel von einer Menge an Knoten ausgegangen, die sich in zwei Partitionen teilen. Nun wird der Cut ermittelt, die Kantenanzahl, die die Grenze der beiden Partitionen schneidet. Dann erfolgt eine paarweise hypothetische Vertauschung von Knoten und nach jeder Vertauschung wird der Cut bestimmt. Jeder Knoten, der bereits bewegt wurde, wird als blockiert markiert und kann nicht erneut getauscht werden. Sind alle Knoten blockiert, so wird der minimale Cut bestimmt. Alle Vertauschungen, die hierfür notwendig sind, werden nun nicht mehr hypothetisch, sondern wirklich durchgeführt. So erhält man am Ende eine Lösung mit minimalem Cut. Der Vorteil, dass die Vertauschungen erst am Ende durchgeführt werden, ist, dass man auch vorübergehende Verschlechterungen akzeptieren kann, um danach eine andere, vielleicht sogar die beste, Lösung zu finden. Dies wird als Hill-Climbing bezeichnet.

Das im folgenden Kapitel beschriebene Verfahren zur Platzierung ähnelt diesem Algorithmus, jedoch gibt es auch einige Unterschiede.

### 5.2.2 Umsetzung

Zunächst sollte man alle benötigten Gatter und ihre Verbindungen untereinander kennen. Nun stellt sich die Frage, wie man die Gatter am sinnvollsten anordnet. Für den Logikteil vor der Scanlogik sollte es nur eine einzige Reihe an Gattern geben, damit darüber noch Platz für die Speicher der Adresse und den TimeStamp bleibt. Da dies ein einfacher Fall ist, werde ich zunächst anhand von diesem die Umsetzung erklären und anschließend auf einen etwas komplexeren Teil, die Scanlogik, eingehen.

Die einzelnen Gatter werden nebeneinander in einer Reihe platziert und anschließend alle Verbindungen eingezeichnet, siehe Abbildung 64.

Nun werden die Plätze der Gatter so lange getauscht, bis man eine günstige Lösung erhält. Diese zeichnet sich dadurch aus, dass es nie mehr als 3 parallele Verbindungen gibt, da aufgrund der Designregeln über die Höhe eines Gatters nur maximal drei Leitungen parallel passen. Es sollte weiter darauf geachtet werden, dass es keine sehr langen Verbindungen gibt,

da diese andere, kürzere, Verbindungen blockieren. Es können entweder alle Möglichkeiten probiert werden wie bei Kernighan –Lin, da dies jedoch sehr aufwendig ist und es viele ähnlich gute Lösungen gibt, wurde hier beim ersten Finden einer guten Lösung aufgehört.

Bei dem Entwurf der Scanlogik kommt zusätzlich eine weitere Möglichkeit hinzu, da hier die logischen Gatter in zwei Reihen platziert werden. Daher muss zusätzlich auf die Anzahl und Möglichkeiten der vertikalen Verbindungen geachtet werden. In Abbildung 64 ist der Entwurf für die Scanlogik zu sehen, die auch genau dies zeigt.

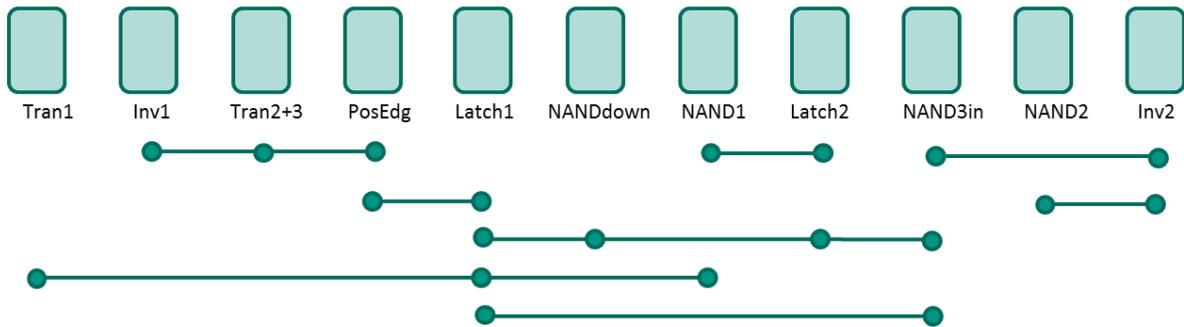


Abbildung 64 Anordnung der Elemente für den digital Teil zwischen Komparator und Scanlogik

### 5.2.3 Beispiele

Im Folgenden werden noch Beispiele des eben beschriebenen Vorgehens gezeigt, die für das Layout des Pixelsensorchips verwendet wurden.

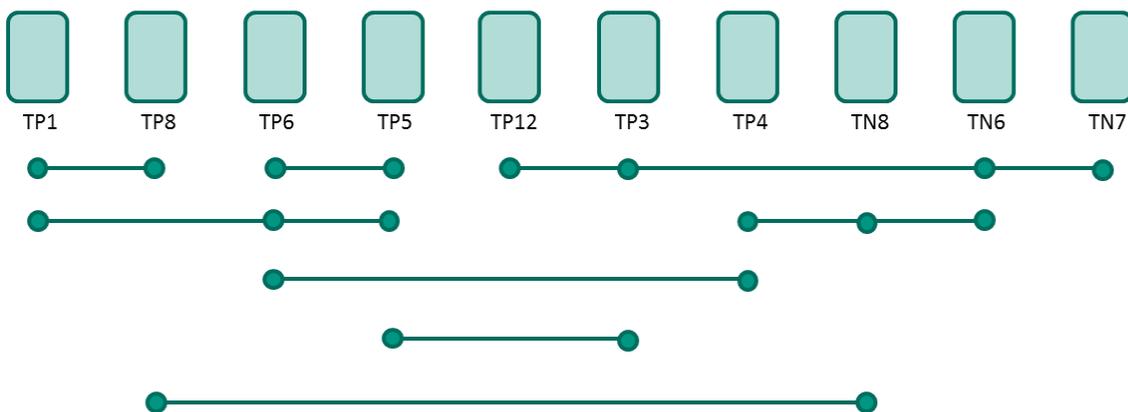


Abbildung 65 Anordnung der Elemente für die Komparatoren mit beiden Modi

Zunächst die Optimierung der Anordnung der Elemente für die beiden Komparatoren und die umgebende Schaltung zur Realisierung der beiden Modi, zu sehen in Abbildung 65. Hier war zusätzlich wichtig, dass die Transistoren TN6 und TN7 sich links am Rand befinden, da sie den Output liefern, der dann weiter in die Logikschaltung aus Abbildung 64 gegeben wird.

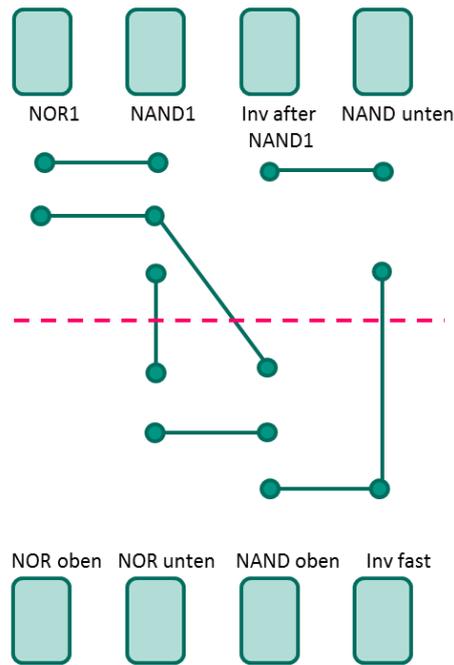


Abbildung 66 Anordnung der Elemente für die Scanlogik

Ein weiteres Element, dessen Anordnung für das Layout untersucht und optimiert wurde, ist die Scanlogik in Abbildung 66. Hier gibt es einen deutlichen Unterschied zu den zuvor gezeigten Anordnungen. Hier sollen die logischen Gatter in zwei Reihen untereinander angeordnet werden, wie in 5.5.1 erklärt wird. Die gestrichelte pinkfarbene Linie soll daher die Trennung der beiden Reihen symbolisieren. Hier muss nun zusätzlich beachtet werden, dass die gestrichelte Linie möglichst selten und wenn, dann einfach, sprich ohne Überkreuzungen und ohne viele diagonale Verbindungen, geschnitten wird.

## 5.3 Element: Komparator

### 5.3.1 Konzept

Jedes Pixelelement benötigt zwei Komparatoren und daher sollte jeder Komparator maximal die Hälfte des für den ADC und den anschließenden Logikteil mit Scanfunktion vorgesehenen Platz in der Breite verbrauchen. So ist es möglich, zwei Komparatoren übereinander zu platzieren und trotzdem nicht zu hoch zu werden. Das Ziel war damit klar: den Komparator so schmal wie möglich zu designen.

### 5.3.2 Umsetzung

In Abbildung 67 ist das Layout des Komparators zu sehen. Es besitzt lediglich eine Breite von  $3,5\mu\text{m}$  und erfüllt damit die Anforderungen. Nun ist es sehr gut möglich, zwei Komparatoren nebeneinander zu platzieren und trotzdem nicht die maximale Breite zu überschreiten. Die Länge beträgt  $14,3\mu\text{m}$ .

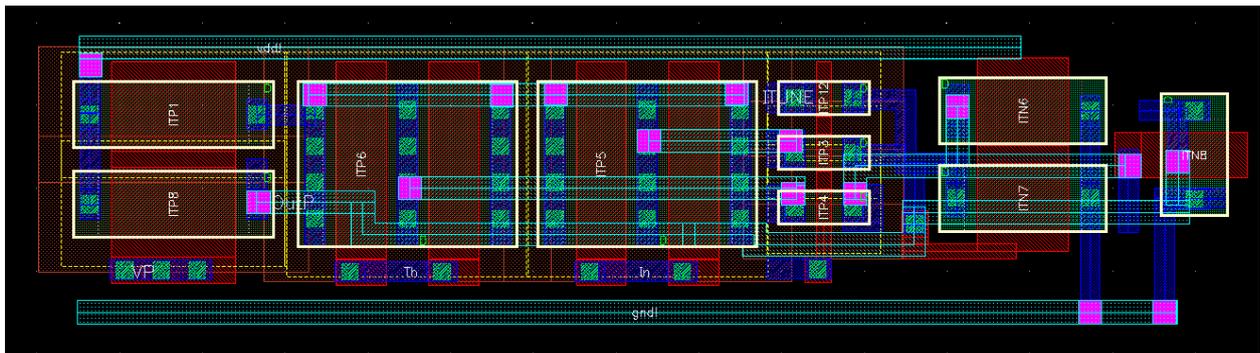


Abbildung 67 Layout Komparator

## 5.4 Element: Logikteil nach dem ADC

### 5.4.1 Konzept

Dieser Teil der Logik sollte sich am besten zwischen den Komparatoren und der Scanlogik befinden, da er als Input den Ausgang der Komparatoren erhält und sein Output als Input der Scanlogik dient. Über diesem Logikteil soll sich außerdem der Speicher für die Adresse und den TimeStamp befinden, daher sollte die Logik nicht zu breit werden, sondern in etwa die Hälfte der zu Verfügung stehenden Breite einnehmen, dies wären ungefähr  $4\mu\text{m}$ . Die Länge sollte  $25\mu\text{m}$  nicht überschreiten.

### 5.4.2 Umsetzung

Hier ist das Layout der Logik zu sehen. Die Platzierung wurde nach dem oben beschriebenen Konzept erstellt und entspricht Abbildung 64. Die Abbildung ist von der Seite zu betrachten. Man erkennt die einzelnen logischen Gatter und ihre Verbindungen deutlich. Das Layout ist  $24,4\mu\text{m}$  lang und  $3\mu\text{m}$  breit.

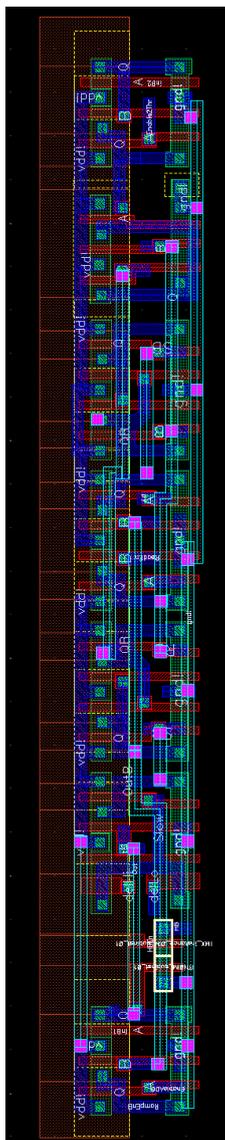


Abbildung 68 Layout digital Teil

## 5.5 Element: Scanlogik

### 5.5.1 Konzept

Die Scanlogik ist ein besonderer Teil des Layouts. Wie in 3.6.1 beschrieben, wiederholt sich das Layout der Scanlogik in Zweierschritten. Dies kommt durch die Struktur der sich abwechselnden NOR und NAND Gatter zustande. Daher teilen sich zwei Elemente (bestehend aus ADC, Logikteil und Speicher) ein Element der Scanlogik. Dieses wird genau in der Mitte zwischen den beiden Elementen platziert. Damit es bei beiden Elementen an der gleichen Stelle liegt, wird das untere Element an der horizontalen Achse gespiegelt. Dadurch grenzt die Scanlogik immer an das Layout des digitalen Teils aus 5.4.

### 5.5.2 Umsetzung

Hier wird in Abbildung 69 das Layout der Scanlogik gezeigt. Deutlich sieht man, dass zwei Reihen an Gattern nebeneinander liegen. Die Scanlogik ist  $7,1\mu\text{m}$  breit und  $7,4\mu\text{m}$  lang. Damit passt sie mit genügend freiem Raum in den vorgesehenen Platz im Layout.

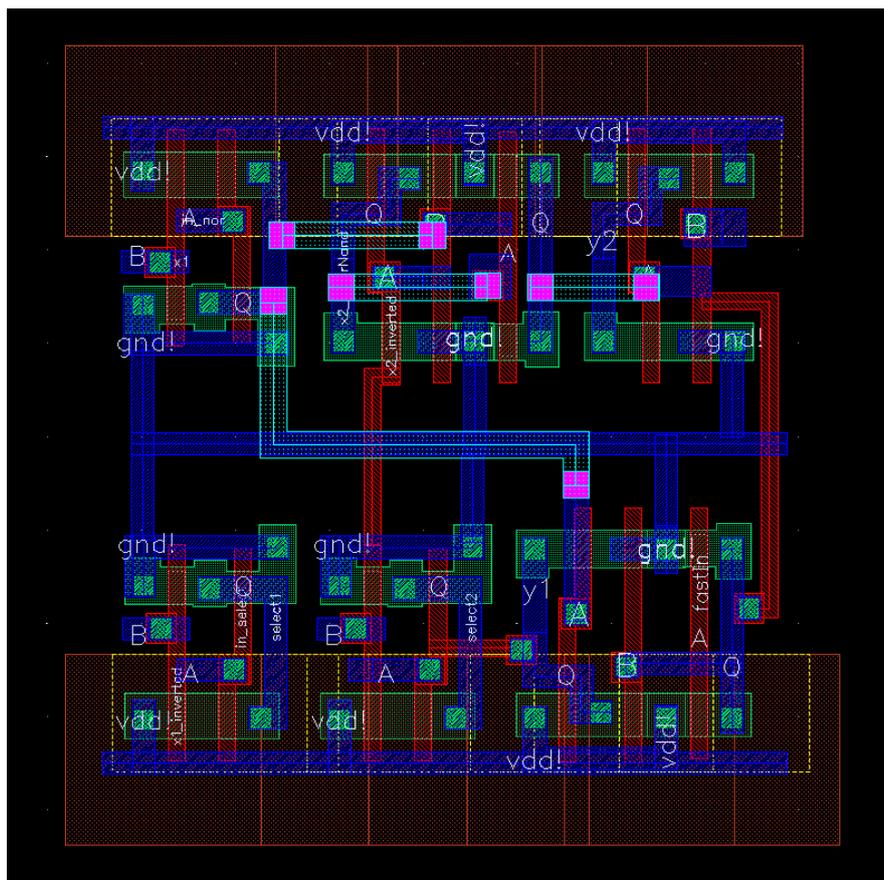


Abbildung 69 Layout Scanlogik

## 5.6 Gesamtschaltung

Anschließend an das Design der einzelnen Elemente werden diese zu den nächst größeren Elementen zusammengefügt und diese erneut zu einem größeren Element, bis man schließlich für das kleinste sich wiederholende Element das Design entworfen hat. Natürlich müssen danach noch die in 3.6 besprochenen Gruppen für die Scanlogik gebildet werden, auch dieses Layout wird kurz vorgestellt.

Zunächst in Abbildung 70 links das Zusammenfügen der einzelnen Elemente ohne die Scanlogik. Es bietet sich eine Betrachtung im Querformat an, die Maße sind  $74,9\mu\text{m}$  auf  $8,6\mu\text{m}$ . Dann sieht man ganz links, bzw. im Hochformat ganz unten, den Speicher, einen RAM. Danach folgen die beiden Komparatoren mit der umgebenden Schaltung für die beiden Modi. Die beiden Komparatoren sind leicht zu finden, da sich das Layout hier wiederholt. Ab hier ist das Layout in zwei Hälften geteilt, eine obere und eine untere. In der oberen Spalte befinden sich die Speicher für den TimeStamp-Wert und für die Adresse. Darunter befindet sich ein großer Logikteil, siehe Abbildung 64, dieser verbindet die Komparatoren mit der Scanlogik. Diese ist im linken Layout nicht zu sehen.

Nun zu dem rechten Layout in der Abbildung. Hier ist das kleinste sich wiederholende Teil gezeigt mit einer Breite von  $17,1\mu\text{m}$  und einer Länge von  $80\mu\text{m}$ . Es besteht aus den zuvor gezeigten Layouts, links in der Abbildung, und der Scanlogik. Die Scanlogik wird von zwei Pixelelementen geteilt, daher diese Gruppierung. Die beiden Pixelelemente wurden an ihrer langen Seite gespiegelt, damit die Scanlogik bei beiden hinter dem Logikteil liegt und die Symmetrie erhalten bleibt.

Das oben vorgestellte Layout wird nun zu einer Gruppe zusammengefügt, dies ist notwendig da sich, wie in 3.6 erklärt, immer die Elemente einer Gruppe das gleiche fastIn teilen. Ein optimales Verhältnis von Gruppenanzahl und Gruppengröße pro Pixel wurde berechnet und es hat sich gezeigt, dass dieses bei 10 Gruppen mit je 20 Elementen pro Spalte liegt. So kann im Mittel die kürzeste Laufzeit erreicht werden. Das Layout einer solchen Gruppe mit genau 20 Elementen, die alle das gleiche fastIn Signal haben, zeigt Abbildung 71. Man sieht sehr schön die sich wiederholenden Elementen aus den beiden kleineren, zuvor gezeigten Layouts. Die Verschaltung der Scanlogik der Elemente wiederholt sich in zweier Schritten. Dies bedeutet keine Änderung der Schaltung, sondern ist eine Optimierung des Layouts. Ebenfalls deutlich sieht man die vertikalen Metallverbindungen der Ebene 3 (hier in Gelb), diese verbinden alle Elemente und verlaufen bis zur Peripherie des Chips. Diese Gruppe ist nun  $80\mu\text{m}$  lang und  $172,2\mu\text{m}$  breit.

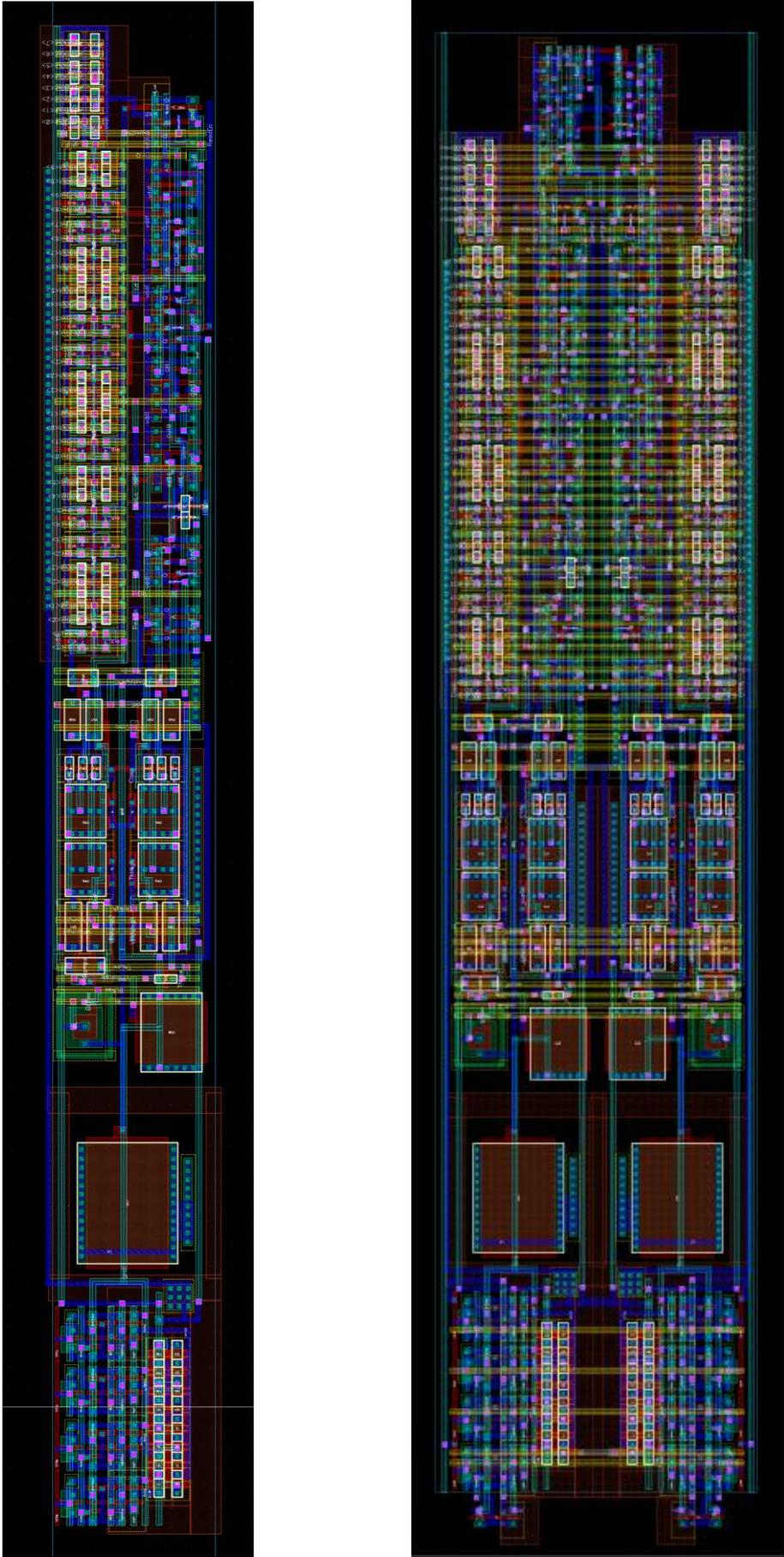


Abbildung 70 Links: Layout der Elemente ohne Scanlogik, rechts: Layout mit Scanlogik

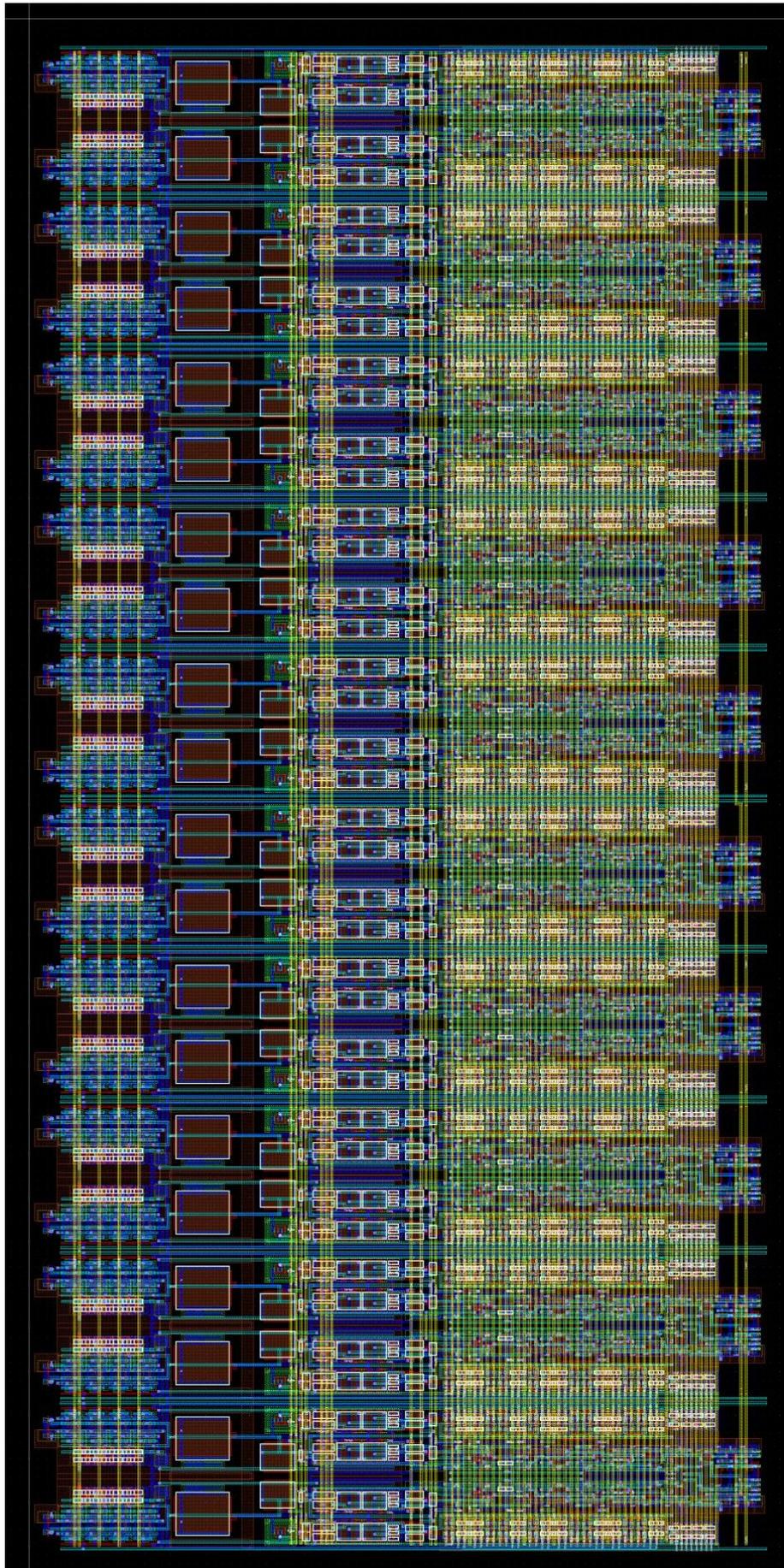


Abbildung 71 Layout einer Gruppe (20 Pixelelemente)

## 6 Anwendung

### 6.1 Mu3e

#### 6.1.1 Hintergründe und Ziel

Ziel des Mu3e Experiments ist es, den Myonenzerfall zu untersuchen. Im Standardmodell gibt es drei Generationen von Leptonen, in jeder Generation gibt es ein geladenes und ein ungeladenes Lepton. Ungeladene Leptonen werden als Neutrino bezeichnet. Am bekanntesten ist die erste Generation, da hier das geladene Lepton das bekannte Elektron ist. In jeder Generation nimmt die Masse der Leptonen zu, damit ist das Elektron das leichteste Lepton. In der zweiten Generation befindet sich als geladenes Lepton das Myon, abgekürzt  $\mu$ . Ziel des Mu3e Experiments ist es, einen Zerfall des Myons in ein Elektron ( $e^-$ ) und zwei Positronen ( $e^+$ ) nachzuweisen. Dieser Zerfall ist laut dem Standardmodell extrem unwahrscheinlich und spielt daher keine besondere Rolle. Es gibt jedoch andere, neue Theorien, die diesen Zerfall erlauben und davon ausgehen, dass dieser mit einer deutlich höheren Wahrscheinlichkeit passiert als im Standardmodell angenommen. Daher wäre der Nachweis des Zerfalls des Myons in  $e^- e^+ e^+$  ein wichtiger Schritt in der Weiterentwicklung des Standardmodells und könnte Hinweise auf die Gültigkeit anderer Theorien, wie beispielsweise der Supersymmetrie, geben. [3]

#### 6.1.2 Aufbau

Das Mu3e Experiment findet am Paul Scherrer Institut in der Schweiz statt. Hier gibt es einen sehr starken Myonen-Strahl. Dieser produziert  $> 10^9$  Myonen pro Sekunde. Diese hohe Dosis hat den Vorteil, dass man damit sehr viele Zufallsprozesse untersuchen kann und so möglichst viele der gewünschten  $e^- e^+ e^+$  Zerfälle beobachten kann.

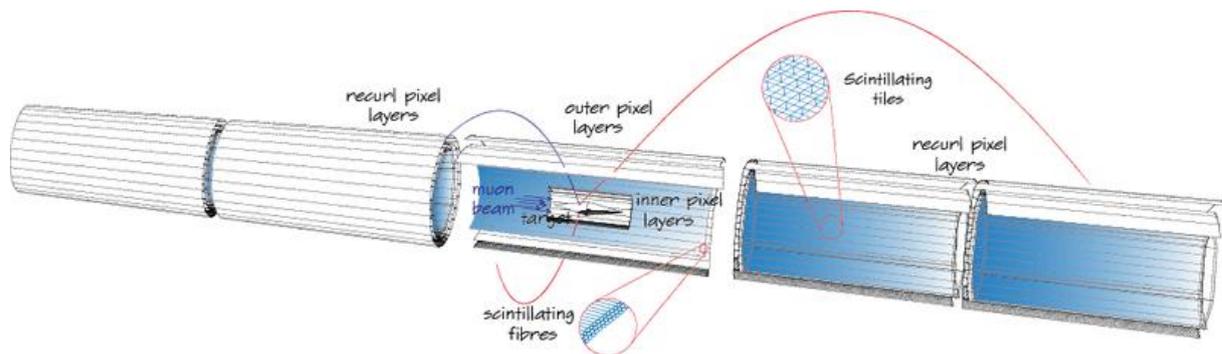


Abbildung 72 Schematischer Aufbau des Mu3e Experiments [3]

Der schematische Aufbau des Mu3e Experiments ist in Abbildung 72 zu sehen. Im Inneren wird der Myonen-Strahl auf das Target geleitet, umgeben zuerst von mehreren Pixellagen. Der in dieser Arbeit entwickelte Pixelsensorchip wird sich in diesen Lagen befinden. Er ist damit ein zentrales Element des Mu3e Experimentes, da mit seiner Hilfe die durch den Zerfall entstehenden Teilchen detektiert werden können. Insgesamt ist der Aufbau von einem hohen Magnetfeld von etwa 1 T umgeben. Dies bewirkt eine Krümmung der Teilchenbahnen, aus

denen sich schließlich die Energien der Teilchen berechnen lassen. Für eine genauere Beschreibung der Realisierung des Aufbaus des Experiments und mehr theoretische Hintergründe sei auf [3] verwiesen.

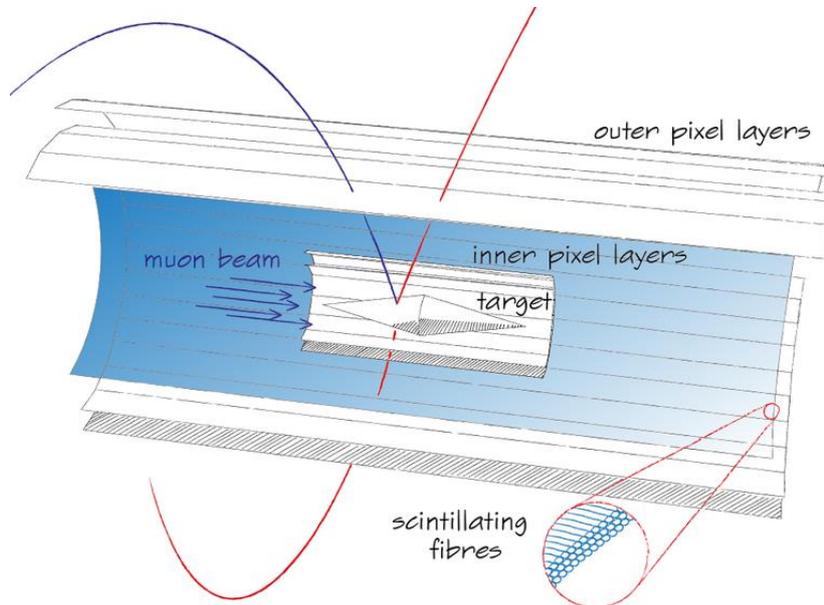


Abbildung 73 Vergrößerte Darstellung des Aufbaus des Mu3e Experiments [3]

## 7 Zusammenfassung und Ausblick

### 7.1 Zusammenfassung

Zusammenfassend wurde in dieser Arbeit ein Pixelsensorchip entworfen, der den Anforderungen des Mu3e Experiments entspricht. In besonderer Weise wurde der analoge und digitale Teil eines Pixelelement neu entworfen und optimiert. Die Entwurfsentscheidungen wurden durch eine Vielzahl von Simulationen bestätigt.

Es wurde das Crosstalkverhalten zwischen dem Amplifier und dem ADC genauer untersucht. Auch wurde der DAC neu entworfen, ab jetzt werden zwei verschiedene Modi unterstützt: einmal der Modus mit zwei unterschiedlich hohen Schwellspannungen, dann der Modus zur Messung der Amplitude des Eingangssignals durch eine linear ansteigende Schwellspannung. Die Steigung dieser Schwellspannung kann von außen gewählt werden. Diese beiden Funktionen gab es bisher für keinen Mu3e Pixelsensorchip. Ebenfalls neu entworfen und optimiert wurde die Scanlogik. Hier konnte durch Optimierung und das neue fastIn eine sehr deutliche zeitliche Optimierung erzielt werden. Hierfür wurden die 200 Pixel einer Spalte in zehn Gruppen mit je 20 Elementen gegliedert. Die Gesamtschaltung wurde simuliert und die Simulationsergebnisse bewertet. Im Anschluss an den Entwurf und die Simulationen wurde das Layout für den analogen und den digitalen Teil des Pixelelements entworfen. Für das Layout ist die Anordnung der einzelnen Gatter und Transistoren zu Beginn von großer Bedeutung, daher wurde speziell für diesen Schritt ein eigenes Optimierungsverfahren entwickelt. Das im Zuge dieser Arbeit entworfene Layout entspricht in besonders guter Weise den Anforderungen, die ein extrem kleines Design fordern. Es wurden alle Elemente auf minimalstem Raum untergebracht, und alle Elemente wurden zu einzelnen Modulen und diese zu größeren Einheiten zusammengefasst, so dass das Layout des gesamten Chips mit wenig Aufwand realisiert werden konnte. Ebenfalls wurde der digitale Teil der Peripherie des Pixelsensorchips untersucht und simuliert. Insgesamt erfüllen alle in dieser Arbeit entworfenen Designs und ihre zugehörigen Layouts alle Voraussetzungen und Anforderungen.

### 7.2 Ausblick

Der im Zuge dieser Arbeit entworfene Pixelsensorchip wird Ende Oktober 2016 dem Hersteller zur Fertigung übergeben. Zwischen dem Zeitpunkt der Übermittlung des Chips und dem Erhalt des fertigen Chips liegen ungefähr zwei bis drei Monate. Daran im Anschluss beginnen die Messungen des Chips. Hier wird geprüft, ob der Entwurf, der bisher durch Simulationen getestet und in seiner Funktionalität bestätigt wurde, auch die erwarteten Ergebnisse bei realen Tests zeigt. Diese Phase wird ungefähr zwei Monate in Anspruch nehmen. Sind alle Tests erfolgreich abgeschlossen, so wird der Pixelsensorchip für das Mu3e Projekt des Paul Scherrer Instituts in der Schweiz eingesetzt werden. Hier sollen im Sommer 2017 die ersten Messungen mit einer starken Neutrinoquelle beginnen. [3]

## Anhang: Literatur

- [1] Lipp, Hans Martin; Becker, Jürgen:  
**Grundlagen der Digitaltechnik**  
7. Auflage, Oldenburg Verlag, München, 2011
- [2] Peric, Ivan:  
**Design and Realisation of Integrated Circuits for the Readout of Pixel Sensors in High-Energy Physics and Biomedical Imaging**  
PhD Thesis, Bonn, 2004
- [3] Paul Scherrer Institut:  
**The Mu3e Experiment**  
<https://www.psi.ch/mu3e/introduction>, abgerufen am 12.10.2016
- [4] Turchetta, Renato; et al.:  
**Analog electronics for radiation detection**  
Taylor & Francis, CRC Press, Boca Raton, Chapter 2, 2016
- [5] Peric, Ivan:  
**Vorlesungsunterlagen zu Design analoger Schaltkreise**  
Karlsruhe Institute of Technology, Karlsruhe, p. 2014
- [6] Peric, Ivan:  
**Vorlesungsunterlagen zu Design digitaler Schaltkreise**  
Karlsruhe Institute of Technology, Karlsruhe, 2015
- [7] Augustin, Heiko et al.:  
**The MuPix system-on-chip for the Mu3e experiment**  
Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 2016
- [8] Peric, Ivan; Takacs, Eszter:  
**Large monolithic particle pixel-detector in high-voltage CMOS technology**  
Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 624, p. 504-508, 2010
- [9] Dössel, Olaf:  
**Vorlesungsskript zu Lineare Elektrische Netze**  
Karlsruhe Institute of Technology, Karlsruhe, 2011
- [10] Becker, Jürgen:  
**Vorlesungsunterlagen zu Hardware Synthese & Optimierung**  
Karlsruhe Institute of Technology, Karlsruhe, 2014

- [11] Peric, Ivan:  
**A novel monolithic pixelated particle detector implemented in high-voltage CMOS technology**  
Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 582, p. 876-885, 2007
- [12] Peric, Ivan et al.:  
**HVCMOS pixel sensors**  
2015 IEEE SENSORS – Proceedings, p. 1-4, 2015
- [13] Peric, Ivan; Kreidl, Christian; Fischer, Peter:  
**Particle pixel detectors in high-voltage CMOS technology—New achievements**  
Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 650, p. 158-162, 2011
- [14] Peric, Ivan; et al.:  
**High-voltage pixel sensors for ATLAS upgrade**  
Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 765, p. 172-176, 2014
- [15] Peric, Ivan; et al.:  
**High-voltage pixel detectors in commercial CMOS technologies for ATLAS, CLIC and Mu3e experiments**  
Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 731, p. 131-136, 2013