

RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG



KIRCHHOFF-INSTITUT FÜR PHYSIK

Fakultät für Physik und Astronomie
Ruprecht-Karls-Universität Heidelberg

Diplomarbeit
im Studiengang Physik

vorgelegt von

Stephan Martens

aus

Göttingen

25. November 2003

Strahlentests mit Mikrochips für das ALICE Experiment

Die Diplomarbeit wurde von Stephan Martens ausgeführt am

Kirchhoff-Institut für Physik

unter der Betreuung von

Prof. Dr. Lindenstruth

Abstract

Im Rahmen dieser Arbeit wurden mehrere Mikrochips bei laufender (monoenergetischer) Protonenbestrahlung auf ihre Funktionsfähigkeit hin überprüft. Zu den überprüften Chips zählen verschiedene Speichertypen (SRAM, SDRAM & FLASH), FPGAs (ACEX1K, APEX20KE & EPXA1) und Prozessoren (EPXA1 & TRAP1). Es wurden Fehlerzahlen bzw. Lebensdauern gemessen und (in den meisten Fällen) auf ihre Flußabhängigkeit untersucht. Daten im FLASH haben sich dabei als sehr strahlenresistent erwiesen, während die Schreibelektronik deutlich früher Schäden davontrug. Bei relativ intensiver Bestrahlung wurden permanente Bitfehler in SDRAM beobachtet. Ansonsten sind keine permanenten Fehler bei den verwendeten Intensitäten und Dosen gefunden worden.

Aus den gemessenen Daten ergeben sich als Worst-Case Abschätzung der MTTF für ein DCS-Board 21 Tage und für einen TRAP1 6 Monate.

Abstract

In the course of this thesis several microchips were tested on their behaviour under (monoenergetic) proton-radiation. Among the measured chips are different memories (SRAM, SDRAM & FLASH), FPGAs (ACEX1K, APEX20KE & EPXA1) and processors EPXA1 & TRAP1). Errorcount respectively lifetime were measured and (in most cases) analysed in terms of fluxdependency. In this connection the data in the FLASH proved to be radiation hard, whereas the writellogic got damaged much earlier. Under quite high radiation permanent bitflips in the SDRAM were observed. No other permanent errors were seen at the used intensities.

From the measured data worst-case estimates for a meantime to failure arise with about 21 days for a DCS-board and about 6 months for a TRAP1.

Inhaltsverzeichnis

1	Einleitung und Motivation	1
1.1	Raumfahrt	1
1.2	Teilchendetektoren	2
1.3	Diese Arbeit	3
2	Aufbau und Funktionsweise der Mikrochips	5
2.1	MOS-FETs und CMOS	5
2.2	SRAM	6
2.3	SDRAM	7
2.4	FLASH	8
2.5	FPGA	9
3	Fehlermodelle	13
3.1	Latchup	13
3.2	Eingeschlossene Ladungen	15
3.3	Kristallschaden	16
3.4	Stoffumwandlung	16
3.5	Direkte Beeinflussung der elektrischen Eigenschaften (Bit-Flips)	17
3.6	Gatedurchbruch	17
4	Die Messungen und ihre Ergebnisse	19
4.1	Aufbau des Experimentes	20
4.2	SRAM	23
4.2.1	Durchführung des Experimentes	23
4.2.2	Die Messergebnisse	24
4.3	SDRAM	28
4.3.1	Durchführung des Experimentes	28
4.3.2	Die Messergebnisse	29
4.3.2.1	Erste Testreihe (memtest3)	29
4.3.2.2	Zweite Testreihe (ohne Betriebssystem)	34
4.3.2.3	Dritte Testreihe (memtest3 auf DCS-Board)	38
4.4	FLASH	40

4.4.1	Durchführung des Experimentes	40
4.4.2	Die Messergebnisse	40
4.4.2.1	Erste Testreihe	40
4.4.2.2	Zweite Testreihe	42
4.4.2.3	Dritte Testreihe	43
4.5	FPGA	44
4.5.1	Durchführung des Experimentes	44
4.5.2	Die Messergebnisse	45
4.6	ARM	49
4.6.1	Durchführung des Experimentes	49
4.6.2	Die Messergebnisse	49
4.7	TRAP	52
4.7.1	Durchführung des Experimentes	52
4.7.2	Die Messergebnisse	52
4.7.2.1	Erste Testreihe	52
4.7.2.2	Zweite Testreihe	53
5	Die Ergebnisse der Flußabhängigkeit Fits	57
6	Schlußfolgerungen	59
6.1	Allgemeines	59
6.2	Berechnung der Wirkungsquerschnitte	59
6.2.1	Wirkungsquerschnitt für lineare Prozesse	59
6.2.2	Wirkungsquerschnitt für quadratische Prozesse	60
6.3	Bestimmung der Zeitkonstanten für Doppeltreffer für SDRAM	61
6.4	Bestimmung der Zeitkonstanten für Doppeltreffer für SRAM	61
6.5	Gemessene Wirkungsquerschnitte	62
6.6	Abschließende Bemerkungen	65
 Anhang		
A	Designs & Software	71
A.1	Kontroll Script	71
A.2	SCSN-Treiber	72
A.2.1	Das Design	72
A.2.2	Das Softwareinterface	73
A.3	Memory-Test Design(s)	73
A.3.1	Das 'mtest' FPGA Design	73
A.3.1.1	'dgen.vhd'	74
A.3.1.2	Der Speicher	74
A.3.1.3	'analyse.vhd'	74

A.3.1.4	'scsn_slave'	74
A.3.1.5	'hash.vhd'	75
A.3.2	'mtest.c'	76
A.4	Work-Test Design(s)	77
A.4.1	Das Work-Test FPGA Design	77
A.4.2	'wtest.c'	78
A.5	ARM-Test Design	79
A.5.1	EPXA1 Design & Software	79
A.5.2	'armlog.c'	80
B	Danksagungen	81
C	Abbildungsverzeichnis	83
D	Akronyme	85
E	Literaturverzeichnis	87

1 Einleitung und Motivation

Es gibt mehrere Gründe, sich mit der Strahlenverträglichkeit von Mikrochips auseinanderzusetzen. Die wichtigsten zivilen hierfür sind wohl (abgesehen von prinzipiellem Wissbegieren) in wahlloser Reihenfolge Raumfahrt und der Einsatz in Teilchendetektoren. Außerdem ist natürlich immer noch zu bedenken, daß jegliche Elektronik wo auch immer sie benutzt wird in leichtem Maße radioaktiver Strahlung ausgesetzt ist, welche z.B. als Höhenstrahlung oder natürliche Aktivität auftritt.

1.1 Raumfahrt

Für die Raumfahrt ist die Strahlenverträglichkeit der Elektronik essentiell, da Elektronik für Satelliten oder interplanetare Sonden in weit höherem Maße radioaktiver Strahlung ausgesetzt ist, als dies auf der Erde der Fall ist. Zwar fallen einerseits terrestrische Quellen wie Höhenstrahlung oder radioaktive Elemente in der Umgebung weg, jedoch fehlen andererseits sowohl einige $\frac{kg}{cm^2}$ Atmosphäre als Abschirmung als auch das Erdmagnetfeld, welches geladene Teilchen zu den Polen ablenkt (letzteres ist für Satelliten noch teilweise vorhanden).

Die Strahlungsquellen außerhalb der Atmosphäre sind vor allem der Sonnenwind, aber auch Überreste von Novae und Supernovae. Diese werden teilweise von interstellaren Wolken weiter auf Energien beschleunigt, die auf der Erde zur Zeit nicht erzeugbar sind. Beim Start von der Erde kommt noch der sogenannte Van Allen Gürtel dazu, in dem vor allem hochenergetische Protonen (100keV[lex] bis mehr als 150MeV[H.V]) und Elektronen (40keV[lex] bis 780keV[H.V]) mit (relativ) hohen Flußdichten vorkommen.

Die Elektronik muß also über mehrere Jahre bis Jahrzehnte mit radioaktiver Strahlung zurechtkommen, ohne in ihrer Funktionsfähigkeit beeinträchtigt zu werden, da zumindest für Sonden keine Möglichkeit der externen Wartung besteht. Für Satelliten ist dies zwar theoretisch möglich, aufgrund der Kosten einer solchen Operation jedoch nicht wirtschaftlich.

1.2 Teilchendetektoren

Bei modernen Hochenergieexperimenten wird ein großer Teil der Elektronik zwangsläufig in den Detektor integriert bzw. wird die Elektronik selbst als Detektor genutzt (z.B. Germanium Detektoren). So werden bei heutigen Experimenten praktisch nur noch direkt elektronisch auslesbare Detektoren benutzt wie z.B. Germanium-Detektoren, Szintillatoren und MWPC¹s. Bei all diesen werden außerdem zumindest noch geeignete Vorverstärker in direkter Nachbarschaft zu den Detektoren benötigt.

Der Motivation dieser Arbeit war das für 2006 am CERN geplante Experiment ALICE² für den LHC³. Aus verschiedenen Gründen wird man dort einen sehr großen Teil der Elektronik in den Detektor mit einbauen, wo sie damit naturgemäß der Strahlung ausgesetzt sein wird. Einer dieser Gründe ist, daß gewisse Triggerentscheidungen sehr schnell getroffen werden müssen (innerhalb weniger μs) um das Detektor-Design handhabbar zu machen. Weitere Gründe sind die gewünschte Empfindlichkeit der Eingänge und die sich aus der Anzahl der Kanäle ergebende Datenflut. Um diese Datenmengen handhaben zu können, will man sie schon möglichst früh (also im Detektor selbst) auf ein erträgliches Maß reduzieren, ohne dabei relevante Daten zu verlieren.

So wird das DCS⁴-Board (welches nicht nur im TRD⁵ sondern auch in der TPC⁶ in großer Zahl Verwendung finden wird) als Einplatinen-Computer mit ARM⁷-Core, SRAM⁸ basiertem FPGA⁹ und SDRAM¹⁰ ausgestattet, und zur Laufzeit des Detektors unter Linux als normaler Computer via Ethernet ansprechbar sein. Insgesamt werden über 750 DCS-Boards im Detektor eingebaut werden. Außerdem werden über 64.000 TRAP¹¹-Chips im TRD verbaut werden[CER01], welche vollständige single-chip-Computer sind. Für das Verständnis des Detektors ist also die Kenntniss der Strahlenverträglichkeit dieser Komponenten essentiell notwendig.

¹„Multi Wire Proportional Chamber“

²„A Large Ion Colider Experiment“

³„Large Hadron Colider“

⁴„Detector Control System“

⁵„Transistion Radiation Detector“

⁶„Time Projection Chamber“

⁷„Advanced RISC Mashine“

⁸„Synchronous RAM“

⁹„Field Programable Gate Array“

¹⁰„Synchronous DRAM“

¹¹„Traclet Processor“

1.3 Diese Arbeit

Um die Strahlenverträglichkeit verschiedener Mikrochips zu verstehen, wurden im Laufe dieser Arbeit FPGA-Designs und Software entwickelt, um ihr Verhalten während laufender Protonenbestrahlung zu beobachten. Diese sollen die im Detektor vorkommenden Protonen und schnellen Neutronen bei weit höheren Flußdichten simulieren. Auf diese Weise soll die radioaktive Komponente der Auswirkungen der gesamten Laufzeit des ALICE Experimentes innerhalb weniger Minuten vorab überprüft werden. So soll eine Abschätzung gewonnen werden, ob das Experiment in der geplanten Form überhaupt durchführbar ist, oder ob man statt normaler COTS¹ auf teure spezial Chips und Designs überwechseln muß.

¹„Components Of The Shelf“

2 Aufbau und Funktionsweise der Mikrochips

Im Folgenden wird kurz der grundlegende Aufbau der Mikrochips und ihre Funktionsweise dargelegt. Dies geschieht in dem Maße, wie es zum Verständnis der Fehlermodelle (Kapitel 3) relevant ist.

2.1 MOS-FETs und CMOS

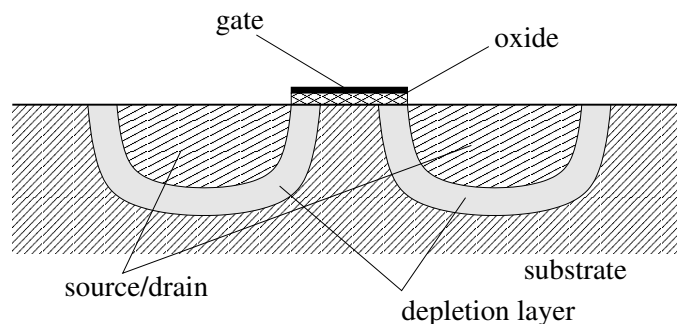


Abbildung 2.1: Typischer Aufbau eines MOS-FET

Der typische Aufbau eines MOS-FET¹, wie er heute eingesetzt wird, ist in Abbildung 2.1 zu sehen. Source und Drain sind hierbei hochdotiert, während das Substrat lediglich schwach in entgegengesetzte Richtung (Donatoren statt Akzeptoren bzw. umgekehrt) dotiert ist. Legt man an das Gate eine geeignete Spannung an, so bildet sich unterhalb der Oxydschicht ein leitfähiger Kanal aus und bildet somit eine leitende Verbindung zwischen Source und Drain. Je nach Dotierung unterscheidet man hierbei p-MOS und n-MOS Transistoren. Die Benennung erfolgt hierbei nach den Majoritätsladungsträgern im leitenden Zustand (welche interessanterweise die Minoritätsladungsträger im Grundzustand sind).

¹„Metal Oxide Semiconductor Field Effect Transistor“

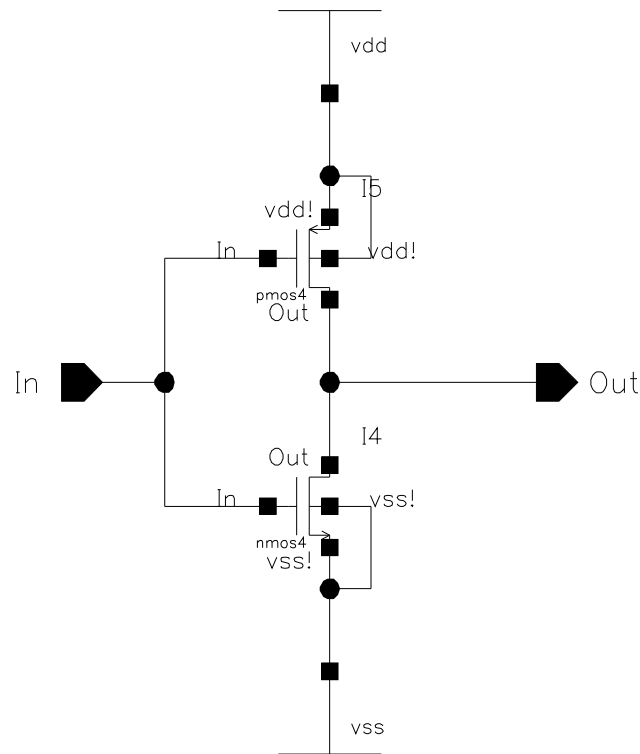


Abbildung 2.2: Aufbau eines CMOS NOT Gatters (push-pull Inverter)

In CMOS¹-Technik werden MOS-FETs beider Art in antisymmetrischem Aufbau betrieben (z.B. NOT Gatter in Abbildung 2.2). Da so bei digitalem Betrieb immer die eine Seite der Schaltung sperrt, fließt abgesehen vom Ausgang praktisch kein Strom. Aus diesem Grunde basieren fast alle in der heutigen Zeit gefertigten digitalen Schaltungen auf diesem Prinzip.

2.2 SRAM

Eine SRAM Zelle besteht aus zwei direkt rückgekoppelten Invertern und zwei Auslesetransistoren, wie in Abbildung 2.3 zu sehen ist. Das Lesen des Bits erfolgt dann über einen Differenzverstärker, der an D und D_n angeschlossen ist. Zum Schreiben werden die Leitungen D und D_n aktiv mit genügend starken Transistoren auf High bzw. Low gezogen, wodurch die Inverter in den entsprechenden Zustand kippen. Da in den Inver-

¹„Complementary Metal Oxide Semiconductor“

tern praktisch kein Querstrom fließt und die Eingänge rein kapazitive Lasten sind, fließt – solange kein Zugriff erfolgt – praktisch kein Strom.

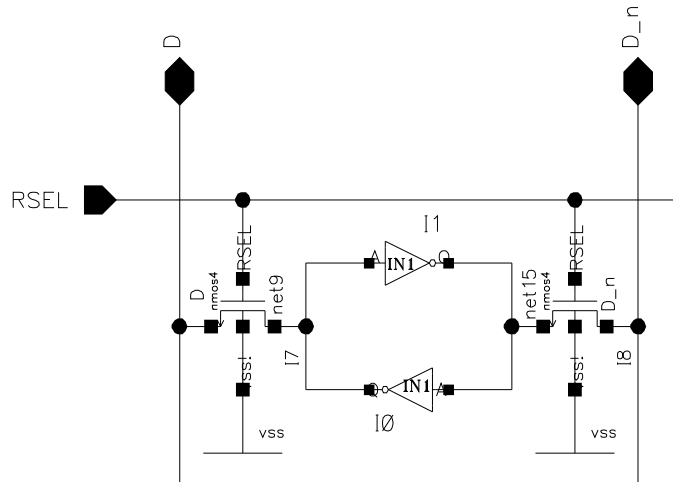


Abbildung 2.3: Aufbau einer SRAM Zelle (ein Lese/Schreib-Port) bestehend aus zwei gegengekoppelten Invertern (siehe Abbildung 2.2) und zwei Auslesetransistoren.

2.3 SDRAM

Eine DRAM¹ Zelle besteht aus einem Kondensator und einem Auslesetransistor, wie in Abbildung 2.4 zu sehen. Der Kondensator – der meist als Sperrschichtkondensator implementiert ist – wird je nach zu speicherndem Bit geladen oder nicht. Mit Hilfe eines Sense Amplifiers kann dann der Wert des Bits bestimmt werden.

Ein Nachteil dieser Methode ist, daß die Daten beim Auslesen verloren gehen, weswegen sie zurückgeschrieben werden müssen (Write-back). Außerdem geht durch Leckströme etc. über die Zeit ein Teil der Ladung verloren. Aus diesem Grunde müssen die Daten in periodischen Abständen ausgelesen und neu geschrieben werden (Refresh).

SDRAMs unterscheiden sich von „normalen“ DRAMs dahingehend, daß die Daten nicht wortweise sondern zeilenweise (typische Größen bewegen sich ohne weiteres im Bereich von 2^{10} Worten) ausgelesen werden und in SRAM oder Latches zwischengepuffert werden. Auf diese Zeilen kann nun viel effizienter zugegriffen werden, da man nicht ein Write-back je Zugriff benötigt. Dies hat jedoch den Nachteil, daß auch für einen Schreibzugriff eine Zeile erst gelesen werden muß.

¹„Dynamic RAM“

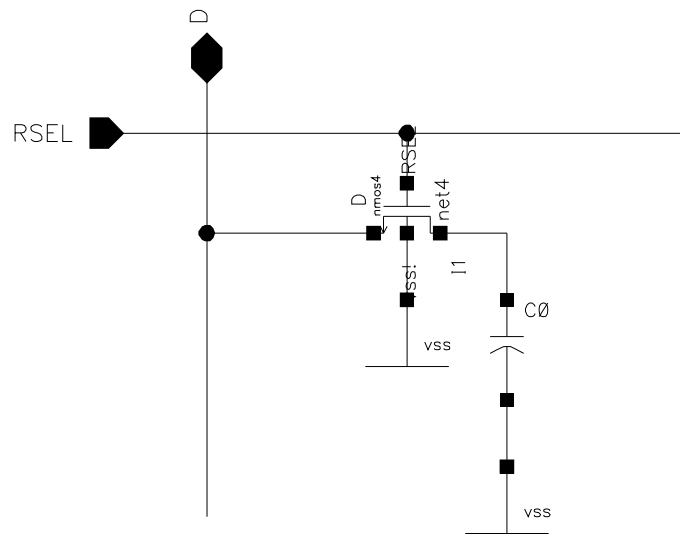


Abbildung 2.4: Aufbau einer DRAM Zelle

Außerdem wird das normalerweise asynchrone Timing der DRAM Chips durch ein synchrones Protokoll ersetzt.

Was den Refresh angeht, so wird heute der größte Teil der Funktionalität im internen Controller der SDRAM Chips versteckt. Von außen sind lediglich die Kommandos zum Durchführen desselben verfügbar, wobei – außer dem direkten Lese/Schreib Zugriff auf bestimmte Zeilen – keinerlei Kontrolle mehr über die zu bearbeitenden Adressen verbleibt.

2.4 FLASH

FLASH EEPROM¹s basieren nicht auf Standard MOS-FET Technologie sondern benötigen eine spezielle Art von Transistoren, wie sie in Abbildung 2.5 zu sehen sind. Durch Anlegen einer genügend hohen Spannung an das Gate läßt sich ein Durchbruch zwischen Kanal und „floating Gate“ erzeugen. Hierdurch gelangen Ladungen auf das Selbige, die den unter ihm liegenden Teil des Kanals beeinflussen. Das Löschen der Daten erfolgt über einen weiteren Anschluß des Transistors, mit dessen Hilfe das „floating gate“ – wieder durch Anlegen einer genügend hohen Spannung – durch einen weiteren Durchbruch entladen werden kann.

¹„Electronically Erasable and Programmable ROM“

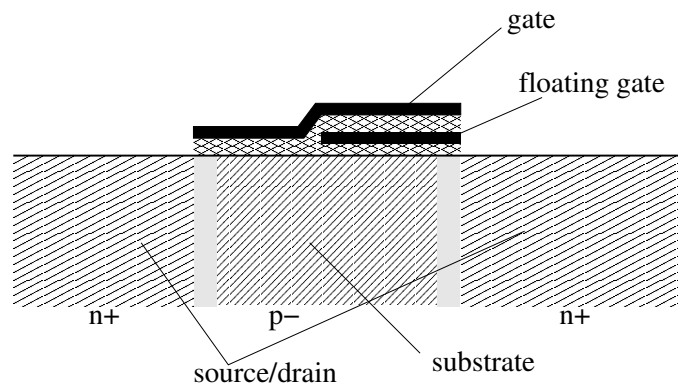


Abbildung 2.5: Aufbau eines FLASH-Transistors

Da Siliziumoxyd unterhalb der Durchbruchfeldstärke ein sehr guter Isolator ist, halten sich die Ladungen in diesem System sehr lange und werden auch durch das Auslesen nicht zerstört. Auch das Programmieren und Löschen zerstört diese Isolationsschicht nicht, da nur sehr wenig Ladung bewegt wird. Trotzdem trägt der FLASH-Speicher nach einigen 100.000 Schreibzyklen permanente Schäden davon, was man auf das Ansammeln von Ladungen im Oxyd oder Degeneration der Kristallstruktur durch zu häufiges Schreiben/Löschen zurückführt. Der vollständige Aufbau einer FLASH-Speicherzelle ist in Abbildung 2.6 zu finden.

Zum Thema FLASH siehe auch [AMD].

2.5 FPGA

FPGAs sind eine Art von frei programmierbaren Logikbausteinen. Zu diesem Zweck gibt es ein Netz von konfigurierbaren Verbindungen welche durch „pass-Transistoren“ geschaltet werden (siehe Abbildung 2.7) Bei den Logik-Blöcken gibt es zwei verschiedene Ausprägungen, die je nach Hersteller und Modell eingesetzt werden. Die einen setzen vollständig auf konfigurierbare „Lookup-Tables“, die anderen setzen zusätzlich noch spezielle Komponenten wie z.B. Addierer oder Multiplizierer ein. Die Logikzellen enthalten außerdem noch eine gewisse Anzahl von Flip-Flops. Abgesehen von den Logikblöcken enthalten viele FPGAs noch eine gewisse Anzahl von Speicherblöcken.

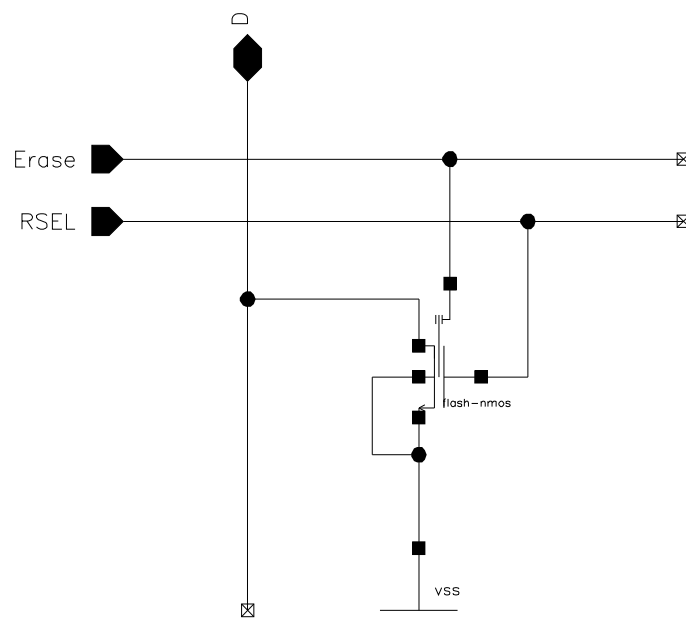


Abbildung 2.6: Aufbau einer FLASH Speicherzelle (Flash-Transistor siehe Abbildung 2.5)

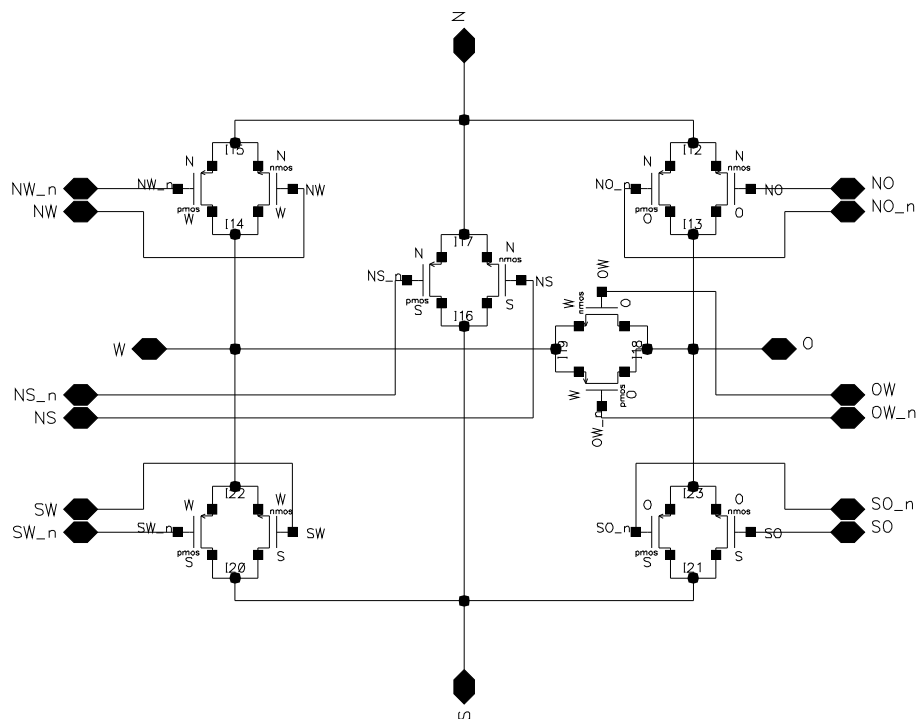


Abbildung 2.7: Kreuzung eines FPGA Verbindungsnetzwerks bestehend aus 6 pass-Transistoren

3 Fehlermodelle

Als mögliche Ursachen für Fehler kommen verschiedene Prozesse in Frage. Die wichtigsten Modelle für derartige Prozesse sind im Folgenden kurz zusammengefasst und erläutert.

Zu tieferen Betrachtungen zu Fehlermodellen und ihren Ursachen siehe z.B. [Fac], [Rho] und [Tip99].

3.1 Latchup

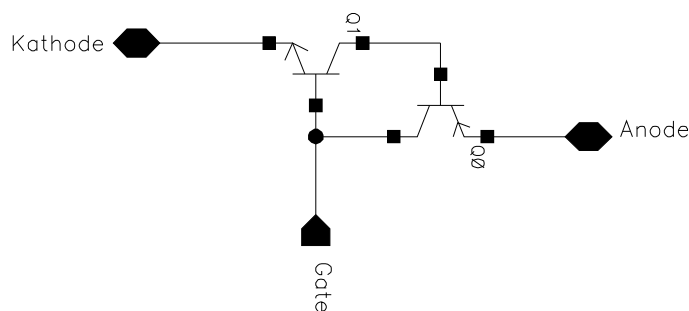


Abbildung 3.1: Aufbau eines Thyristors aus zwei Bipolar-Transistoren

Beim Latchup handelt es sich indirekt um eine permanente Schädigung. Der permanente Schaden wird in diesem Falle durch eine Überhitzung verursacht, die durch einen Kurzschluß zwischen V_{dd} und V_{ss} hervorgerufen wird. Dieser Kurzschluß wird im Falle des Latchups durch einen parasitären Thyristor (Thyristor siehe Abbildung 3.1) hergestellt. Bei geeigneter Kontrolle der Stromzufuhr (Strombegrenzung) muß dies allerdings nicht zu einer permanenten Schädigung führen.

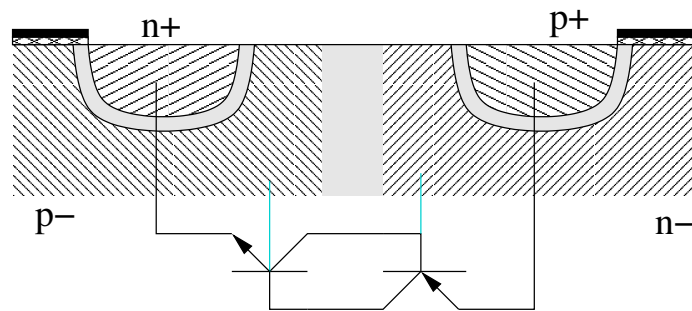


Abbildung 3.2: Zustandekommen parasitärer Thyristoren durch Nutzung von p-MOS und n-MOS Transistoren (Thyristorschaltplan eingezeichnet)

Solche parasitären Thyristoren (siehe Abbildung 3.2) treten ungewollt in jeder CMOS-Schaltung auf, die nicht auf einem SOI¹-Prozeß beruht. Hierbei ist entweder n- oder p- – je nach benutztem Prozess – eine größere Wanne, in der die entsprechenden Transistoren untergebracht werden. Die Gefahr des Latchups besteht nicht nur bei Bestrahlung der Mikrochips, sondern ist ein allgemeines Problem, das man mit verschiedenen Designtechniken zu verhindern sucht. Eine dieser Techniken ist der Einbau von sogenannten „Guard-Rings“, die für eine gute Verbindung des Substrates (Bulk-Anschluß der Transistoren) zu den entsprechenden Potentialen sorgen, sodaß der Steuereingang des Thyristors definiert auf „Aus“ liegt.

Ionisierende Strahlung verstärkt die Gefahr des Latchup, da durch sie freie Ladungsträger in die Sperrschichten zwischen den schwach dotierten Regionen eingebracht werden. Der durch das anliegende Feld hervorgerufene Strom kann nun unter Umständen diese parasitären Thyristoren trotzdem zünden.

Um dies zu Vermeiden bieten einige Hersteller zusätzlich zu den normalen Prozessen noch weitere an, die nach eigenen Angaben maskenkompatibel sind, jedoch auf speziellen epi²-Wafeln gefertigt werden (siehe z.B. [BAE]). Im Gegensatz zu den normalen Wafeln wird hier die Dicke der niedrig dotierten Schicht kleiner gehalten. Eine darunter liegende höher dotierte Schicht dient vor allem zum Verkleinern der Verarmungszonen welche als aktives Volumen für die Ladungssammlung dienen. Zusätzlich wird damit der Substratwiderstand gesenkt und der Anschluß an V_{ss} (bzw. V_{dd} je nach Prozess) verbessert.

Eine weitere Methode zur Verhinderung des Latchups ist die SOI-Technologie, in der die Transistoren auf einen Isolator aufgebracht werden und meist auch zwischen den verschiedenen Transistoren eine Isolationsschicht plaziert wird. Dies wird mit Hilfe von implantiertem Sauerstoff erreicht, der sich mit dem Silizium zu SiO_2 verbindet. Dies verhindert nicht nur das Auftreten parasitärer Thyristoren, sondern hat noch einige weite-

¹„Silicon On Insulator“

²„epitactical“

re Vorteile wie z.B. eine drastische Reduktion der Leckströme und daraus resultierender Störeffekte.

Der Ort der Schädigung ist beim Latchup nicht zwingend in direkter Nachbarschaft zum Thyristor selbst, da z.B. auch die stromführende Leitung (z.B. der Bondingdraht) durch den hervorgerufenen Stromfluß durchbrennen kann. Außerdem können durch die Überhitzung auch weitere benachbarte Gebiete in Mitleidenschaft gezogen werden. Meist wird sogar der gesamte Chip – und nicht nur ein Logikpfad – hierdurch zerstört.

3.2 Eingeschlossene Ladungen

Die ionisierende Strahlung erzeugt bei der Durchquerung des Materials Elektron-Loch Paare. Diese Paare werden jedoch nicht nur im dotierten Bereich gebildet, wo sie nach gewisser Zeit rekombinieren, sondern auch in den isolierenden Oxydschichten oder – einen entsprechenden Prozess vorausgesetzt – im undotierten Silizium. Diese Paare rekombinieren zwar größtenteils wieder, aber aufgrund der unterschiedlichen Beweglichkeiten von Löchern und Elektronen und einer noch größeren Unsymmetrie in ihren Tunnelwahrscheinlichkeiten können sich Löcher im Isolator nahe der leitenden Bereiche ansammeln. Dies geschieht in den Übergangsbereichen zwischen den Kristallgittern des (dotierten) Siliziums und des Siliziumdioxids, da dort die Bindungen unter Spannung stehen. Diese Ladungen haben – wenn sie sich im Gate-Oxyd befinden – effektiv gesehen einen direkten Einfluß auf die Schwellenspannung des Transistors, da sie wie Ladungen auf dem Gate wirken. Im Feldoxyd erhöhen sie lediglich die Leckströme.

Die Anfälligkeit für diese Art von Strahlenschäden steigt im Gegensatz zu den meisten anderen beim Annealing an, da durch das Ausheilen von Defekten die Anzahl der Fallenplätze für die Löcher zunimmt.

Elektronen können zwar nicht effektiv im Oxyd eingeschlossen werden, jedoch gibt es einen vergleichbaren Prozess, der auch negative Ladungen im Oxyd speichern kann.

Bei diesen Prozessen handelt es sich zwar um kumulative Effekte – d.h. sie werden mit zunehmender Bestrahlung stärker – jedoch strebt die Störung gegen einen Grenzwert.

Genauer zu diesen Prozessen siehe z.B. [Tip99]. In digitaler CMOS-Logik sind diese Effekte jedoch größtenteils tolerierbar, da eine geringe Abweichung in den Schwellenspannungen auf das digitale Verhalten kaum einen Einfluß hat.

3.3 Kristallschaden

Bei Bestrahlung von Materie mit hochenergetischen Ionen wird nicht die gesamte Energie in Form von Ionisation umgewandelt, sondern es kommt auch zum NIEL¹. Dieser Energieübertrag kann z.B. zur Bildung von Defekten (Frenkel-Defekte) im Kristallgitter führen. Diese Frenkel-Defekte beeinflussen nun auf zwei Arten die elektrischen Eigenschaften der Mikrochips.

Erstens bilden sie Streuzentren für die Elektronen resp. Löcher. Hierdurch wird nicht nur ihre Beweglichkeit verringert, sondern diese Störstellen dienen außerdem noch als mögliche Rekombinationspunkte wodurch zusätzlich noch die Lebensdauer der freien (vor allem der Minoritäts-) Ladungsträger reduziert wird.

Zweitens wirken diese Defekte als Dotierung und wandeln n-dotiertes Material in p-dotiertes Material um. Genaueres hierzu siehe z.B. [Rho].

Im Falle von digitaler Logik (besonders CMOS) ist der Effekt allerdings relativ klein, da nur ein geringer Teil der Energie als NIEL abgegeben wird, und eine geringe Verschiebung der Schwellenspannungen und der fließenden Ströme ohne Probleme toleriert werden können.

Es handelt sich hierbei jedoch um kumulative Effekte, die aber mit der Zeit wieder heilen können. Die Halbwertszeiten dieser Heilung sind jedoch im Allgemeinen bei Raumtemperatur sehr groß. Durch Erhitzen auf mehrere 100°C lässt sich dieser Prozess jedoch u.U. auf wenige Tage verkürzen.

3.4 Stoffumwandlung

Eine Stoffumwandlung kann auf zweierlei Art erfolgen. Die „einfachere“ Methode ist der Einfang eines langsamen Neutrons oder Protons. Diese Umwandlung ist allerdings hauptsächlich für niederenergetische Neutronen relevant und auch hier ist der Wirkungsquerschnitt sehr gering und dürfte bei den zu erwartenden Stromdichten vernachlässigbar ausfallen, auch wenn die Mikrochips nach der Bestrahlung leicht radioaktiv aktiviert waren. Auch die zweite Methode der Stoffumwandlung – die Kernspaltung – ist für die verwendeten Materialien vernachlässigbar, da andere Prozesse vorher schon den Chip zerstören.

Beide Effekte ändern die effektive Dotierung und im Falle der Kernspaltung wird zusätzlich noch ein Kristalldefekt erzeugt, der weitere Effekte nach sich zieht (siehe 3.3).

¹ „Non Ionising Energy Loss“

3.5 Direkte Beeinflussung der elektrischen Eigenschaften (Bit-Flips)

Trifft ein ionisierendes Teilchen eine im Betrieb befindliche Schaltung, so kann die deponierte Ladung auch direkt auf deren Ausgabe überkoppeln. Dies kann beispielsweise passieren, indem im aktiven Bereich eines nicht geschalteten Transistors genügend Ladung deponiert wird, um einen Stromfluß zu erzeugen, der den Ausgang dieser Stufe so beeinflusst, daß ein falsches Ergebnis erzeugt wird. Genauso kann, wenn – wie z.B. bei DRAM – ein Kondensator verwendet wird, dieser kurzzeitig durch die eingebrachten Ladungen mit V_{ss} bzw. V_{dd} verbunden werden. Hierdurch kann sich die gespeicherte Ladung signifikant ändern. Außerdem besteht auch die Möglichkeit, daß so unabhängige Bereiche (z.B. die Drains verschiedener Transistoren) über das Substrat zeitweise kurzgeschlossen werden.

Passiert dieser Treffer in einem rückgekoppelten System, wie z.B. einer SRAM-Zelle (siehe Abschnitt 2.2), so kann dieser Treffer den gespeicherten Wert permanent ändern. Diese Art von „Bit-Flip“ ist jedoch nur in dem Sinne permanent, daß das entsprechende Bit verlorengegangen ist. Sein Wert kann – durch Redundanz oder ähnliches – ohne weiteres auf den korrekten Wert restauriert werden. So ist zwar die Software respektive die Daten (mehr oder weniger) permanent geschädigt, die Hardware jedoch noch intakt.

Tritt dieser Fehler nicht in einem direkt rückgekoppelten System auf, so wird er unter Umständen nicht wahrgenommen, da die Schaltung das korrekte Ergebnis rechtzeitig aus den Eingangsdaten rekonstruiert, bevor der Fehler in einer Speicherzelle gespeichert wird.

3.6 Gatedurchbruch

Werden auf einem Chip mehrere verschiedene Spannungen verwendet, und ist eine dieser Spannungen größer als die Durchbruchspannung des Gates im Chip verwendeter Transistoren, so kann es passieren, daß durch Einwirkungen der Strahlung diese Spannung an das Gate gelegt wird (siehe auch Abschnitt 3.5) wodurch dieses permanent geschädigt wird (permanenter Kurzschluß zum Kanal). Dies ist heutzutage nicht mehr nur bei „Hochspannungs“-Bauteilen wie Spannungsreglern und ähnlichem möglich, sondern aufgrund der immer weiter sinkenden Strukturgrößen ein generelles Problem. Dies liegt daran, daß die im Chip verwendeten Spannungen weiter abnehmen müssen, während für die IO¹-Spannungen andere Bedingungen gelten. Dies führt dazu, daß schon heute häufig Kernspannung (Größenordnung 1,3V - 1,8V bei heutigen Prozessoren) und IO-Spannung (1.8V - 5V) verschiedene Werte haben. Die Transistoren im Kern vertragen jedoch derartige Spannungen häufig nicht mehr.

¹„Input/Output“

3 Fehlermodelle

Auch in FLASH-Bausteinen werden verschiedene Spannungen zum Programmieren und Auslesen der Zellen verwendet. Inwieweit die verschiedenen Transistoren in diesem Fall mit den verschiedenen Spannungen zurechtkommen hängt nun vom Design ab.

4 Die Messungen und ihre Ergebnisse

Die Mikrochips wurden in insgesamt drei Strahlzeiten am OCL¹ in Oslo mit 28.5 MeV Protonen bestrahlt. Der verwendete Strahl hatte einen quadratischen Querschnitt von 1cm^2 (Abgesehen von einem kurzen Testlauf mit einem Querschnitt von 0.49cm^2). Dies führte dazu, daß nicht alle Mikrochips komplett bestrahlt werden konnten. So ist es den Messergebnissen des SDRAMs deutlich anzusehen, daß dies nicht der Fall war (siehe 4.3). Bei einem nachträglichen Öffnen des Gehäuses stellte sich heraus, daß der SDRAM Chip aus einem etwa 15mm langen Stück Silizium bestand. Bei dem verwendeten Querschnitt war ein vollständiges Bestrahlen also unmöglich. Auch bei den anderen Mikrochips (vor allem den FPGAs) besteht die Möglichkeit, daß sie nicht vollständig bestrahlt wurden. Es steht jedoch zu vermuten, daß dies (in erster Näherung) lediglich eine Korrektur um den Faktor $\frac{\text{Gesamtfläche}}{\text{Strahlfläche}}$ (bzw. seinem Kehrwert) bedeutet.

Der Strahlstrom schwankte bei den benutzten Intensitäten um bis zu 50% innerhalb einiger Sekunden (bei 10pA mit Karte). Hierzu kommt noch eine Ungenauigkeit in der Wiederholbarkeit der Stromstärke zwischen einzelnen Läufen. Wurde der Strom ohne Karte gemessen, dann mit Karte eine Messung gemacht und der Strom ohne Karte kontrolliert, so ergaben sich teilweise sehr große Abweichungen (bis zu etwa Faktor 2). Dies geschah, obwohl die Schwankungen während der Messungen deutlich kleiner als das waren. Ob diese Ungenauigkeit in der Strommessung oder in der Extraktion der Protonen aus der Ionenquelle begründet liegt, ist nicht bekannt. Allerdings ist die Extraktion die wahrscheinlichere Fehlerquelle.

Eine weitere **systematische** Ungenauigkeit – die erst nach den ersten Meßreihen entdeckt wurde – ist, daß der Strahlstrom, welcher *mit* Karte im Strahl gemessen wurde meist *größer* war als ohne! Es kommen also aus irgendeinem Grunde effektiv gesehen positive Ladungsträger zum Strahl dazu. Als mögliche Erklärung hierfür wurde von Jon Wikne (OCL) (dem Operator des Zyklotrons) das Herausschlagen von Protonen aus dem Material des Boards vorgeschlagen. Die Messungen sind – solange nicht anders angegeben –

¹„Oslo Cyclotron Laboratory“ Universitetet i Oslo

alle auf die Strommessung mit der Karte im Strahl normiert.

Messungen mit verschiedenen Energien waren am OCL nicht praktikabel. Die Ursache hierfür liegt im Beschleuniger, der größere Änderungen der Einstellungen gefordert hätte. Die einzige Alternative wäre das Abbremsen der Protonen in Materie gewesen. Dies hätte jedoch noch weitere unschöne Nebeneffekte gehabt (Genauigkeit, erhöhte Streuung usw.). Aus diesem Grunde wurde auf derartige Messungen verzichtet.

Die Messungen bei verschiedenen Flußdichten waren ursprünglich dazu gedacht bei niedrigen Stromdichten das Verhalten zu verstehen und durch die hohen Stromdichten die kumulativen Effekte zu beschleunigen. Erwartet wurde ein lineares Verhalten (Fehler-rate \sim Flußdichte, bzw. Lebensdauer $\sim \frac{1}{\text{Flußdichte}}$). Ein solches lineares Verhalten ließe auf eine konstante Fehlerwahrscheinlichkeit je eingestrahlttem Teilchen schließen. Andere Exponenten sind Hinweise auf Koinzidenzen (Fehlerrate \sim Flußdichte^c entspricht c-facher Koinzidenz). Beispiele für derartige Prozesse sind u.a. in der dopplerfreien Anregung von Atomniveaus zu finden, die mit zwei Photonenprozessen arbeitet.

4.1 Aufbau des Experimentes

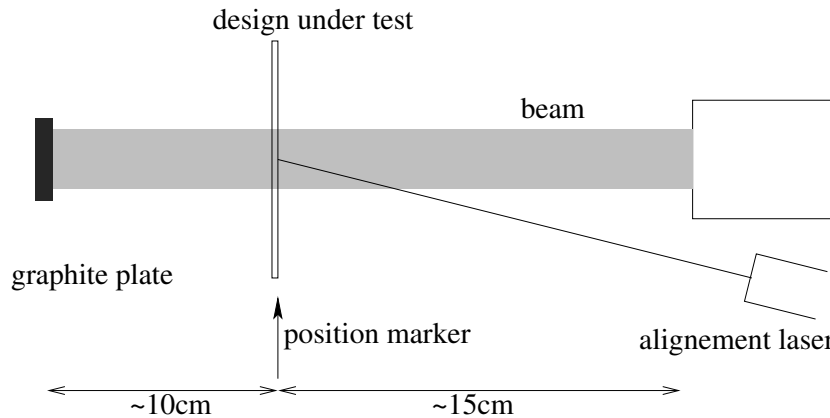


Abbildung 4.1: Aufbau des Experimentes bei der ersten und zweiten Strahlzeit

Der Aufbau des Experimentes bei den ersten beiden Strahlzeiten ist in Abbildung 4.1 zu erkennen. Mit Hilfe einer Quarzscheibe (Statt der Chips) und eines Lasers konnte der Chip so plaziert werden, daß man hoffen konnte, das ganze Silizium zu treffen. Bei der dritten Strahlzeit wurde ein leicht abgewandelter Aufbau (Abbildung 4.2) verwendet, um eine genauere Positionierung der Mikrochips – durch die Vermeidung der Paralaxe – zu ermöglichen. Der Spiegel mußte für die Tests selbst aus dem Strahlengang entfernt

werden. Er wurde nur zum Justieren des Lasers und der Mikrochips montiert. Die Graphitscheibe wurde zum Monitoren des Strahlstroms benutzt.

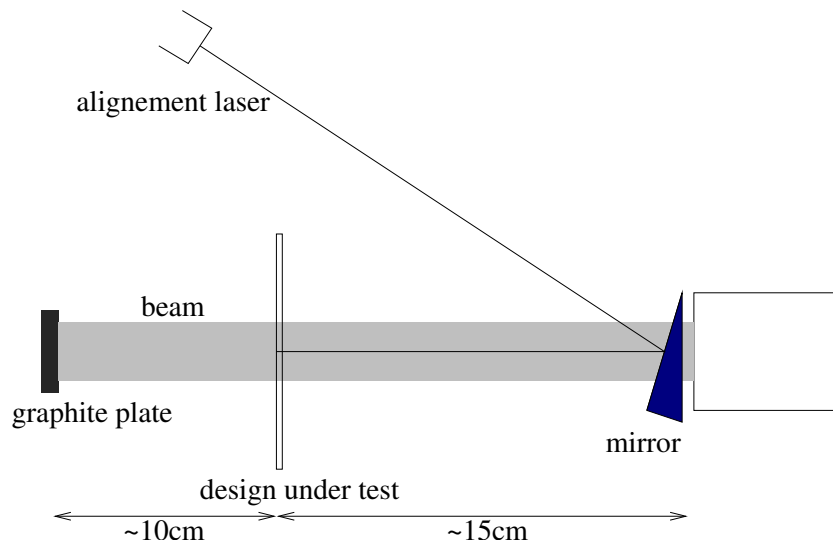


Abbildung 4.2: Aufbau des Experiments bei der dritten Strahlzeit

Das Fixieren der Chips war immer eine größere Aktion, da die zur Verfügung stehenden Boards nicht für eine derartige Anwendung gedacht waren. Als Beispiel hierfür siehe Abbildung 4.3.

Zu den benutzten FPGA-Designs, Software und der Datenübertragung siehe Anhang A.

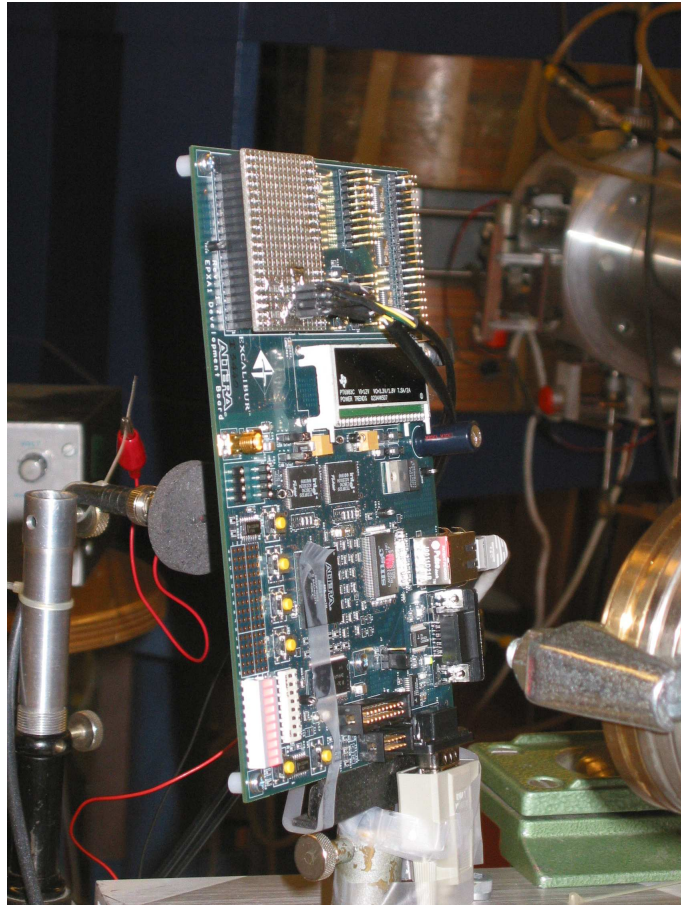


Abbildung 4.3: Aufbau in Natura. Links ist die Graphitscheibe zum Monitoren des Strahlstroms, rechts der Strahlauslass und im Hintergrund ein Ulenkmagnet für ein anderes Target zu sehen. (Vergleiche Abbildung 4.1)

4.2 SRAM

Es wurden SRAM-Tests mit zwei verschiedenen Chips durchgeführt:

- ein „IDT 71V35781“ auf einer ACEX-Karte[Ang01]
- ein „MT 58L512L18F -85A“ auf einer CIA-RORC-Karte[Ata]

Hierbei wurde am IDT-Chip nur die prinzipielle Realisierbarkeit des Experimentes getestet und inwieweit permanente Schädigungen auftreten. Aus diesem Grunde wurden auch weit weniger Messungen mit ihm durchgeführt.

Die Auslese aus dem Speicher erfolgte kontinuierlich und die Ergebnisse wurden zum Computer übertragen (Polling). Hierbei wurde die Fehlerzahl eines Durchlaufs u.U. auch mehr als einmal gelesen.

Weiteres siehe Anhang A.3.

Die Messungen beziehen sich auf 2^{16} Worte mit je 16 bits.

4.2.1 Durchführung des Experimentes

Das Experiment wurde wie folgt durchgeführt:

1. Der „alignment laser“ wurde mit Hilfe der Quarzscheibe ausgerichtet.
2. Die Karte wurde mit dem zu untersuchenden Chip als Ziel montiert.
3. Die Karte wurde programmiert und ihre Funktionsfähigkeit überprüft.
4. Die Auslese wurde gestartet (dies impliziert ein Neuprogrammieren des Systems).
5. Der Strahl wurde angeschaltet und auf Nennwert gebracht.
6. Nachdem genügend Fehler akumulierte waren, wurden Strahl und Programm gestoppt.

Zwischen den Durchläufen wurde die Karte nicht bewegt.

4.2.2 Die Messergebnisse

In den im folgenden besprochenen Abbildungen (Fehlerzahl gegen Zeit) ist als Unsicherheit für die Fehlerzahl die statistisch zu erwartende Fluktuation von \sqrt{N} eingetragen. Für die Fits jedoch, welche differentiell vorgenommen wurden – d.h. es wurde nur der Verlauf zur Zeit der Bestrahlung berücksichtigt ohne Ausnutzung von Anfangs- und Endzeit – wurde eine konstante Unsicherheit vorausgesetzt. Die Genauigkeit der Zeitmessung ist besser als eine Sekunde. Die Zeitwerte sind jedoch nur auf eine Sekunde genau gespeichert, weswegen sich hier ein Fehler von $\pm \frac{1}{2}s$ ergibt (nicht eingetragen).

Es wurden nur dann Daten gespeichert, wenn entweder ein anderer Fehlerwert als zuvor gemessen wurde oder eine Minute vergangen war (siehe hierzu auch Anhang A.3)! Die daraus entstehenden Unsicherheiten im Fit sollten sich jedoch rausheben, da die Messungen einen ausreichend linearen Verlauf zeigen.

Für die Stromdichte ist in den Graphiken die mittlere Schwankungsbreite als Fehler angegeben.

Der Hauptgrund weswegen die Fits differentiell gemacht wurden ist, daß der Strahlstrom zu Beginn der Messung immer sehr ungenau war und auf den Nennwert eingeregelt werden mußte.

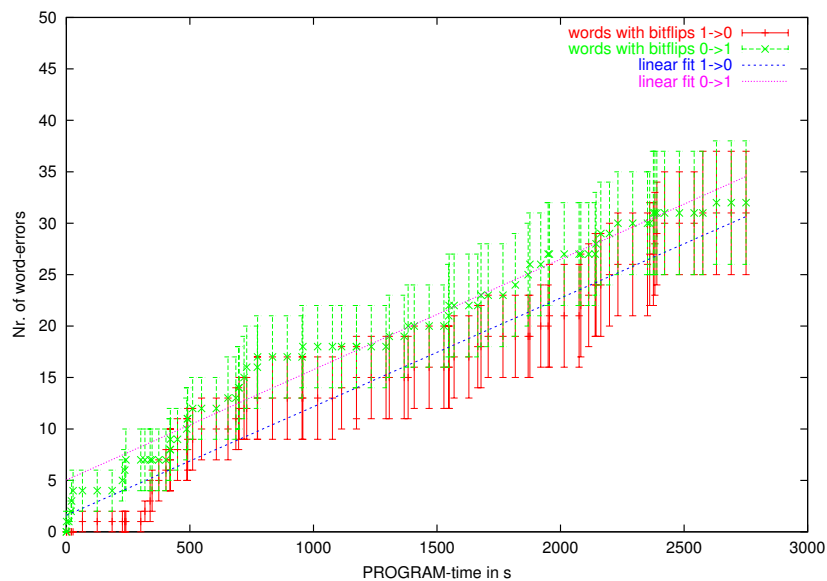


Abbildung 4.4: Fehlerzahl gegen Programm-Laufzeit in MT-SRAM bei $10 \frac{pA}{cm^2}$

Die Abweichungen vom linearen Verlauf bei $10 \frac{pA}{cm^2}$ (Abbildung 4.4) am Anfang (kleiner $\sim 100s$) wurden durch das Einstellen des Strahls verursacht. Die Knicke bei $\sim 300s$ und $\sim 700s$ beruhen vermutlich auf der in 4 angesprochenen Strahlinstabilität.

Der benutzte Fitbereich der Geraden ist 100s-2750s.

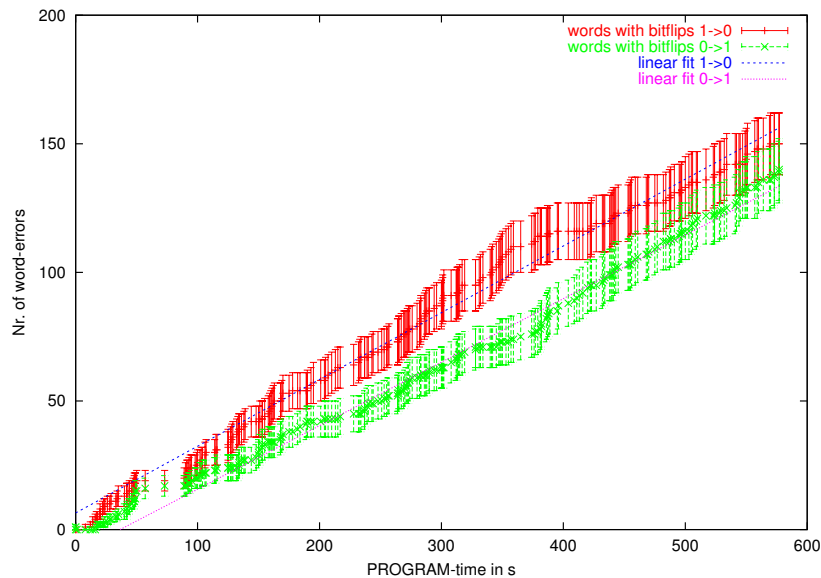


Abbildung 4.5: Fehlerzahl gegen Programm-Laufzeit in MT-SRAM bei $50 \frac{\text{pA}}{\text{cm}^2}$

Auch bei $50 \frac{\text{pA}}{\text{cm}^2}$ (Abbildung 4.5) erkennt man bei ~80s Probleme mit dem Strahl. Die hier vorliegende Unterbrechung des Fehleranstiegs liegt meines Erachtens in einem kurzen Strahlausfall begründet, weswegen der Fitbereich hier bei 100s-575s liegt. Interessant ist die Lücke zwischen den beiden Kurven bei ~375s, welches man nicht sinnvoll erklären kann. Vermutliche Ursache ist eine statistische Fluktuation.

Der Fitbereich bei $100 \frac{\text{pA}}{\text{cm}^2}$ (Abbildung 4.6) ist 20s-600s. Diese Messung ist (vermutlich auch aufgrund der hohen Flußdichte) sehr homogen und linear.

Trägt man die Steigungen der gefitteten Geraden gegen den Strom auf, so ergibt sich Abbildung 4.7. Man erkennt gut, daß der quadratische Fit die beste Approximation an die Meßwerte ist.

Achtung: Es wurde nicht ein Polynom zweiten Grades gefittet, sondern $f_\alpha(x) = \alpha x^2$. Außerdem erkennt man, daß die Fehlerrichtung (bitflips $1 \rightarrow 0$ bzw. $0 \rightarrow 1$) praktisch keine Rolle spielt, weswegen auch nur jeweils eine Kurve je Typ eingetragen ist.

4 Die Messungen und ihre Ergebnisse

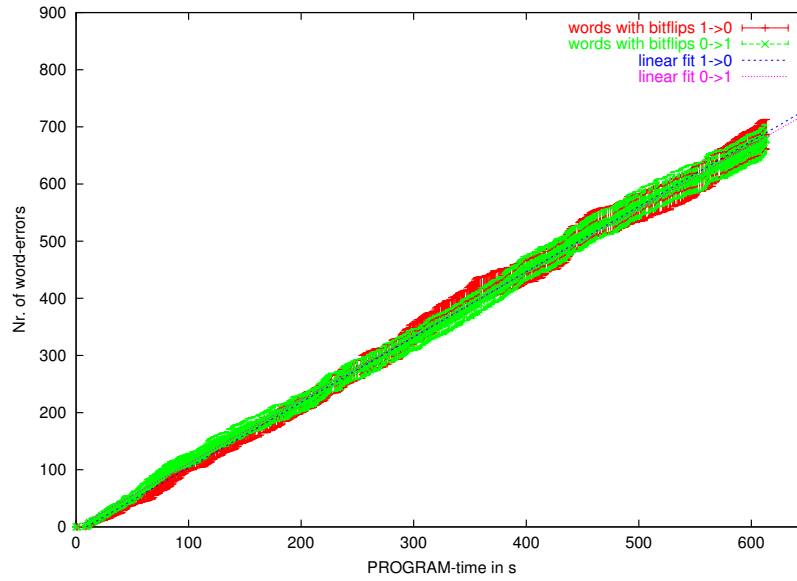


Abbildung 4.6: Fehlerzahl gegen Programm-Laufzeit in MT-SRAM bei $100 \frac{\text{pA}}{\text{cm}^2}$

Fitergebnisse Abbildung 4.7

Funktion	Parameter
αx^2	$\alpha = 1.122 * 10^{-4} \frac{\frac{\text{Errors}}{\text{s}}}{\left(\frac{\text{pA}}{\text{cm}^2}\right)^2} \pm 1.1\%$
ax	$a = 8.0 * 10^{-3} \frac{\frac{\text{Errors}}{\text{s}}}{\frac{\text{pA}}{\text{cm}^2}} \pm 25\%$
$mx + b$	$m = 1.17 * 10^{-2} \frac{\frac{\text{Errors}}{\text{s}}}{\frac{\text{pA}}{\text{cm}^2}} \pm 9\%$
	$b = -0.11 \frac{\text{Errors}}{\text{s}} \pm 18\%$

Oberhalb von $100 \frac{\text{pA}}{\text{cm}^2}$ wurde versucht den IDT-SRAM auch bei $300 \frac{\text{pA}}{\text{cm}^2}$ und $500 \frac{\text{pA}}{\text{cm}^2}$ zum Laufen zu bekommen. Der Chip funktionierte bei diesen Flußdichten jedoch nicht mehr. Er wurde durch diesen Test jedoch nicht beschädigt.

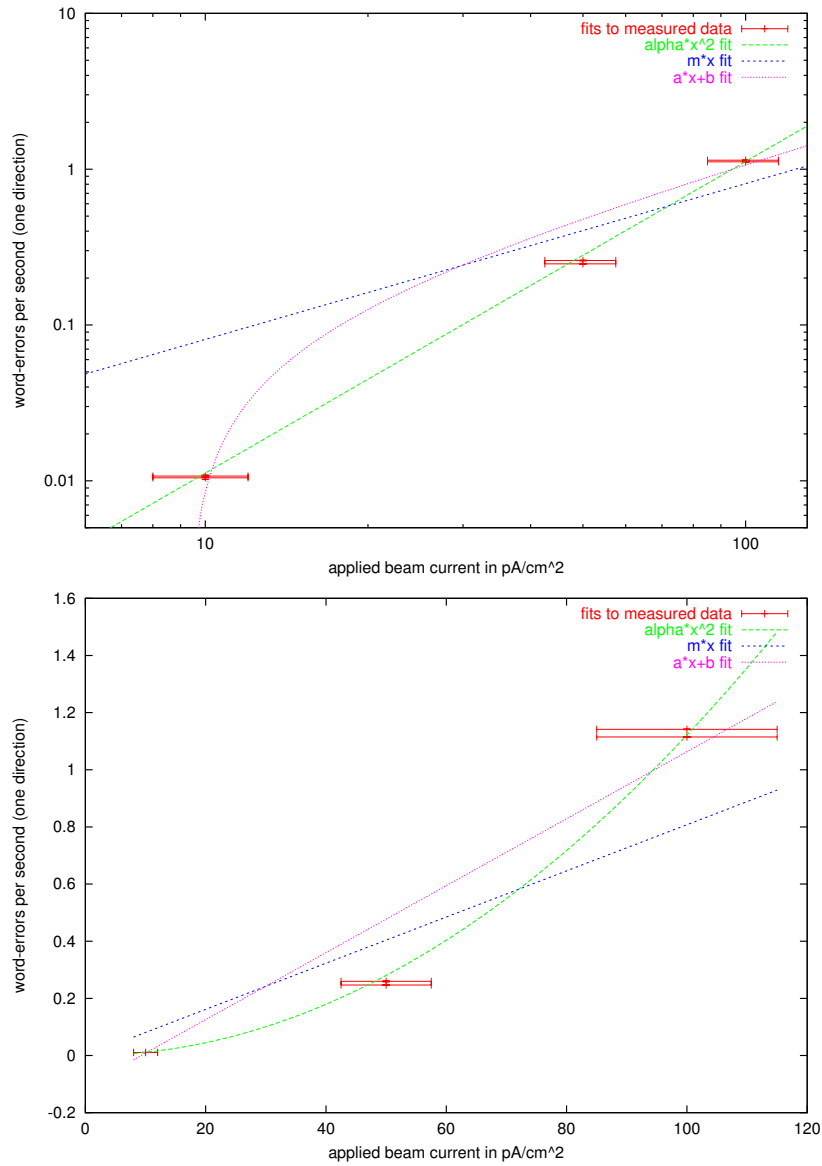


Abbildung 4.7: Flußabhängigkeit der Fehlerrate in MT-SRAM in doppelt logarithmischer und linearer Darstellung

4.3 SDRAM

Getestet wurden folgende SDRAM Chip:

- „MT 48LC16M16A2 -7E B“ auf einem EPXA1-Development-Board[ALTa]
- „MT 48LC16M16A2 -7E C“ auf drei verschiedenen DCS-Boards

Diese SDRAM-Chips bestehen intern aus 4 unabhängigen Bänken, die über denselben Bus angesteuert werden. Jede Bank besteht aus 8k Zeilen mit je 512 16 Bit breiten Worten, also 1024 Byte je Zeile[Mic].

Weiteres zur verwendeten Software und ihrer Funktionsweise siehe [Pan] und [Til].

4.3.1 Durchführung des Experimentes

Der „MT 48LC16M16A2 -7E B“ wurde zweimal getestet.

Beim ersten Experiment wurde ein Memory-Test Programm memtest3 von Ralf Panse[Pan] verwendet, welches unter Linux auf dem ARM-Core des EPXA1 lief. Dies führt allerdings zu dem Problem, daß sowohl die zu testenden Daten als auch das Programm im selben SDRAM-Chip lagen. Auch lassen sich in den Logfiles einige Linux-Fehlermeldungen finden. Ihre Ursache sind strahlungsinduzierte Bitflips in der Page-Table und anderer Linux-Systembereiche, da das Betriebssystem im selben SDRAM Chip lief. In der zwischen den Strahlzeiten zur Verfügung stehenden Zeit war es jedoch nicht möglich, ein Programm zu schreiben, das auf dem ARM-Core – ohne Linux – rein im EPXA1 internen SRAM läuft, da schon ein leeres C-Programm die Größe des SRAMs (insgesamt 32kB[ALTD]) um etwa 100% überstieg. Ursache sind die dazugelinkten Standardbibliotheken.

Die Durchführung erfolgte folgendermaßen:

1. Der „alignment laser“ wurde mit Hilfe der Quarzscheibe ausgerichtet.
2. Die Karte wurde mit dem zu untersuchenden Chip als Ziel montiert.
3. Die Karte wurde aus dem FLASH mit Linux gebootet.
4. Das Testprogramm wurde via NFS¹ von einem Server gemountet
5. Das Programm wurde gestartet.
6. Der Strahl wurde angeschaltet und auf Nennwert gebracht.

¹„Network File System“

7. Nach einer gewissen Zeit wurde der Strahl abgeschaltet und kurze Zeit später auch das Testprogramm (sofern es noch lief).

Zwischen den Läufen wurde die Karte nicht bewegt!

Im Gegensatz zum SRAM wurden hierbei die Werte allerdings während der Bestrahlung laufend neu geschrieben!

Das zweite Experiment wurde von Dr. Heinz Tilsner (KIP¹) durchgeführt. Hierbei lief kein Betriebssystem sondern nur das Testprogramm. *Die Strommessung ist auf den Wert ohne Karte normiert.*

Beim dritten Mal wurden (in drei Durchgängen) drei SDRAM-Chips – diesmal der „MT 48LC16M16A2 -7E C“ – auf jeweils einem DCS-Boards bestrahlt. Die Durchführung war wie beim ersten Test (s.o.) mit dem Unterschied, daß Programm und Strahl gleichzeitig ausgeschaltet wurden, und daß die Werte auf eine *Strommessung ohne Karte normiert* sind.

Getestet wurden jeweils 16MB des Chips also 2^{24} Worte a 8 Bit, bzw. 2^{23} Worte a 16 Bit. Jedoch wurden zumindest beim ersten Test vermutlich lediglich etwa die Hälfte davon bestrahlt wie man aus den Abbildungen 4.12 und 4.14 ersehen kann. Bei den anderen beiden Durchläufen ist eine derartige Aussage nicht möglich, da die Karten zwischen den Durchläufen bewegt wurden.

4.3.2 Die Messergebnisse

4.3.2.1 Erste Testreihe (memtest3)

In den Abbildungen 4.8, 4.9 und 4.10 ist die Fehlerzahl je Durchlauf (konstante Zeit) jeweils in einfach logarithmischer Darstellung dargestellt.

Je „Run“ wurden hintereinander vier Durchläufe mit jeweils einem anderen Testpattern durchgeführt. Diese waren (in dieser Reihenfolge): 0xAA 0xFF 0x00 und 0x55. Diese sind in den Graphiken nebeneinander dargestellt.

Im Normalfall ist nur eine moderate Anzahl von Fehlern je Testdurchlauf zu verzeichnen. Jedoch in einigen wenigen Fällen ist die Fehlerzahl sehr groß. Betrachtet man die Verteilung der Fehler im Adressraum genauer (Aufgetragen gegen die Nr. des Durchlaufs), so erhält man die Abbildungen 4.11, 4.12, 4.13 und 4.14. Aufgrund der geringen Fehlerzahl macht die entsprechende Darstellung für $25 \frac{pA}{cm^2}$ wenig Sinn.

4 Die Messungen und ihre Ergebnisse

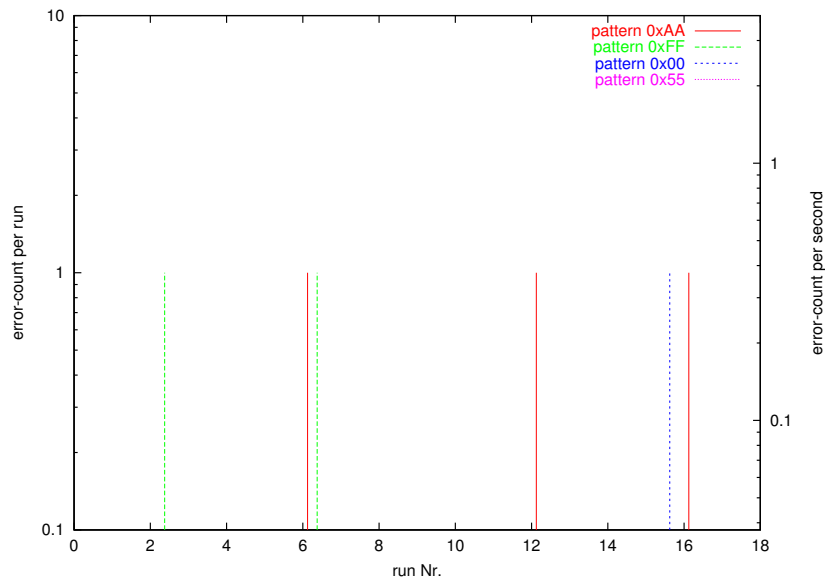


Abbildung 4.8: Fehler je Durchlauf in SDRAM bei $25 \frac{\mu\text{A}}{\text{cm}^2}$

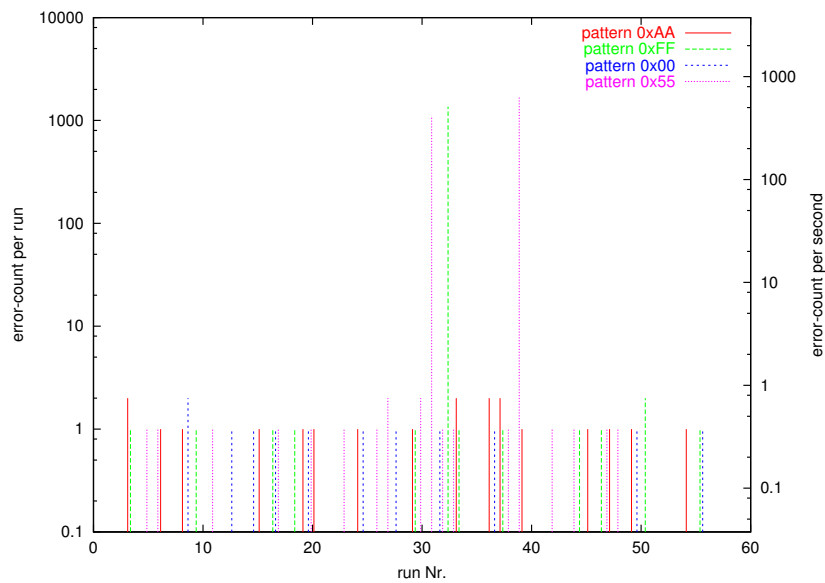


Abbildung 4.9: Fehler je Durchlauf in SDRAM bei $50 \frac{\mu\text{A}}{\text{cm}^2}$

Man erkennt in den Abbildungen 4.12 und 4.14, daß die „Block“-Fehler alles andere als gleichmäßig über den gesamten Bereich verteilt sind. Im Gegenteil, die Fehler sind sehr

¹„Kirchhoff Institut für Physik“ Universität Heidelberg

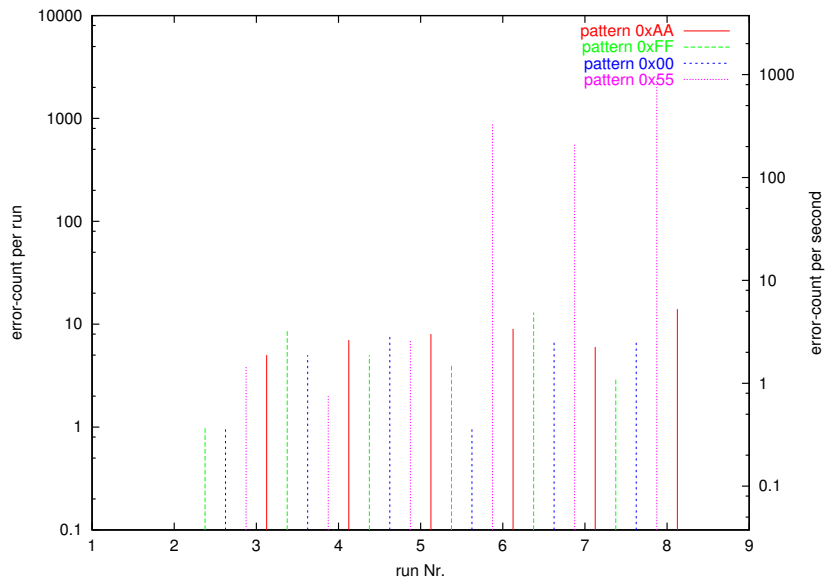


Abbildung 4.10: Fehler je Durchlauf in SDRAM bei $225 \frac{pA}{cm^2}$

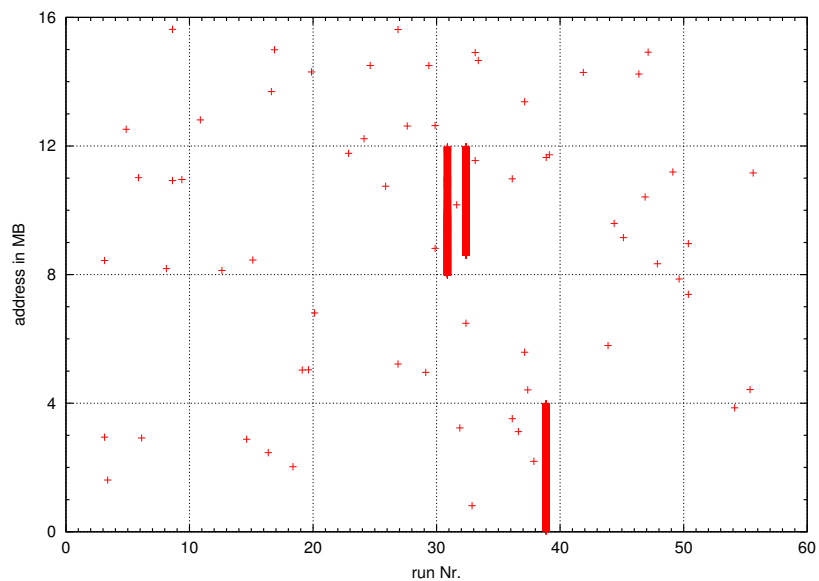


Abbildung 4.11: Fehlerpositionen in SDRAM bei $50 \frac{pA}{cm^2}$

stark auf bestimmte Spalten konzentriert. Außerdem ist eindeutig nur etwa die Hälfte des Chips bestrahlt worden, da praktisch keinerlei Fehler in den Bänken 0 und 1 (wenn man von einer entsprechenden Abbildung in den Adressraum ausgeht) auftreten.

4 Die Messungen und ihre Ergebnisse

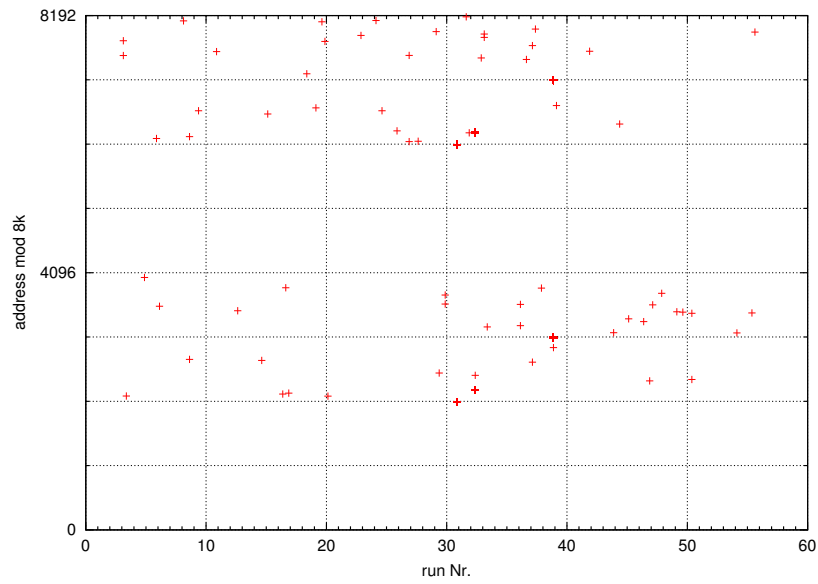


Abbildung 4.12: Fehlerpositionen in SDRAM modulo 8k bei $50 \frac{\text{pA}}{\text{cm}^2}$

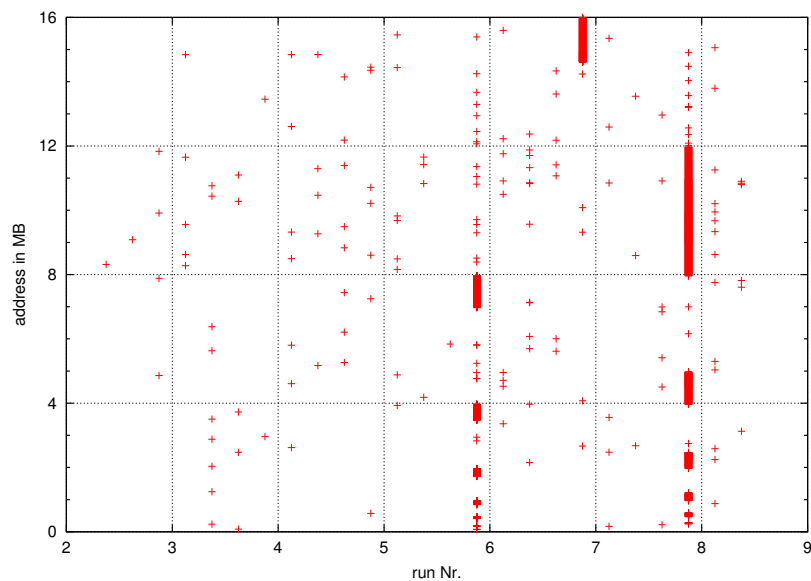


Abbildung 4.13: Fehlerpositionen in SDRAM bei $225 \frac{\text{pA}}{\text{cm}^2}$

Trägt man die Messergebnisse gegen die Flußdichte auf, ergibt sich Abbildung 4.15 für die Einzelfehler.

Auch hier zeigt sich wieder, daß die quadratische Approximation die beste ist. Für den Fit

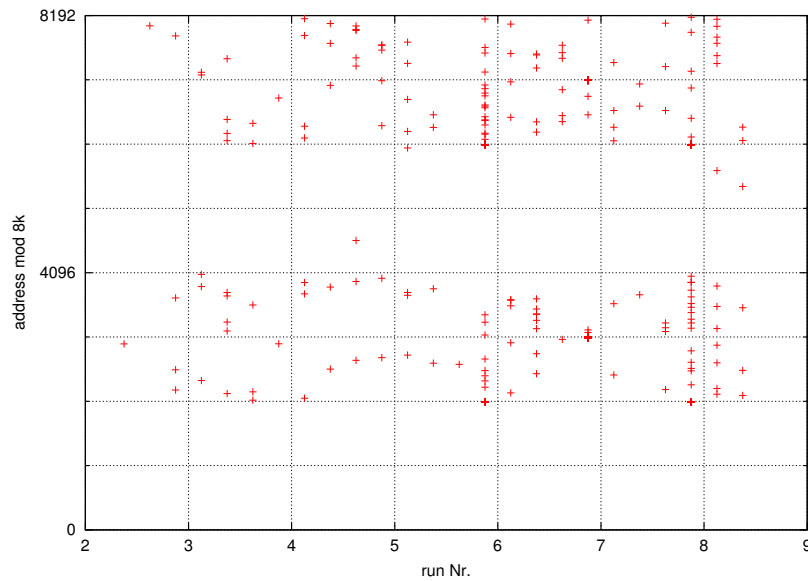


Abbildung 4.14: Fehlerpositionen in SDRAM modulo 8k bei $225 \frac{\mu A}{cm^2}$

werden alle vier Werte gleich behandelt, auch wenn sie wie man sieht nicht alle innerhalb der eigenen Fehlerbereiche liegen und systematische Abweichungen zu beobachten sind. Jedoch zeigen sie alle denselben Verlauf (innerhalb der Fehlergrenzen). Allerdings ist es nicht zu verstehen, weswegen es signifikante Unterschiede zwischen den Test-Mustern 0xAA und 0x55 geben sollte, da beide in ihrer Binärdarstellung abwechselnd aus gesetzten und gelöschten Bits bestehen.

Fitergebnisse Abbildung 4.15

Funktion	Parameter
αx^2	$\alpha = 4.2 * 10^{-5} \frac{\frac{Errors}{s}}{\left(\frac{\mu A}{cm^2}\right)^2} \pm 8\%$
$m x$	$m = 2.7 * 10^{-3} \frac{\frac{Errors}{s}}{\frac{\mu A}{cm^2}} \pm 26\%$
$a x + b$	$a = 6.3 * 10^{-3} \frac{\frac{Errors}{s}}{\frac{\mu A}{cm^2}} \pm 26\%$
	$b = -0.18 \frac{Errors}{s} \pm 42\%$

Beim Auftragen der „Block“-Fehlerraten ergibt sich Abbildung 4.16. Hier läßt sich keine Aussage treffen, ob die Fehlerrate linear oder quadratisch von der Flußdichte abhängt.

Fitergebnisse Abbildung 4.16

Funktion	Parameter
αx^2	$\alpha = 3.0 * 10^{-6} \frac{\frac{Errors}{s}}{\left(\frac{\mu A}{cm^2}\right)^2} \pm 28\%$
$m x$	$m = 3.2 * 10^{-4} \frac{\frac{Errors}{s}}{\frac{\mu A}{cm^2}} \pm 28\%$

4 Die Messungen und ihre Ergebnisse

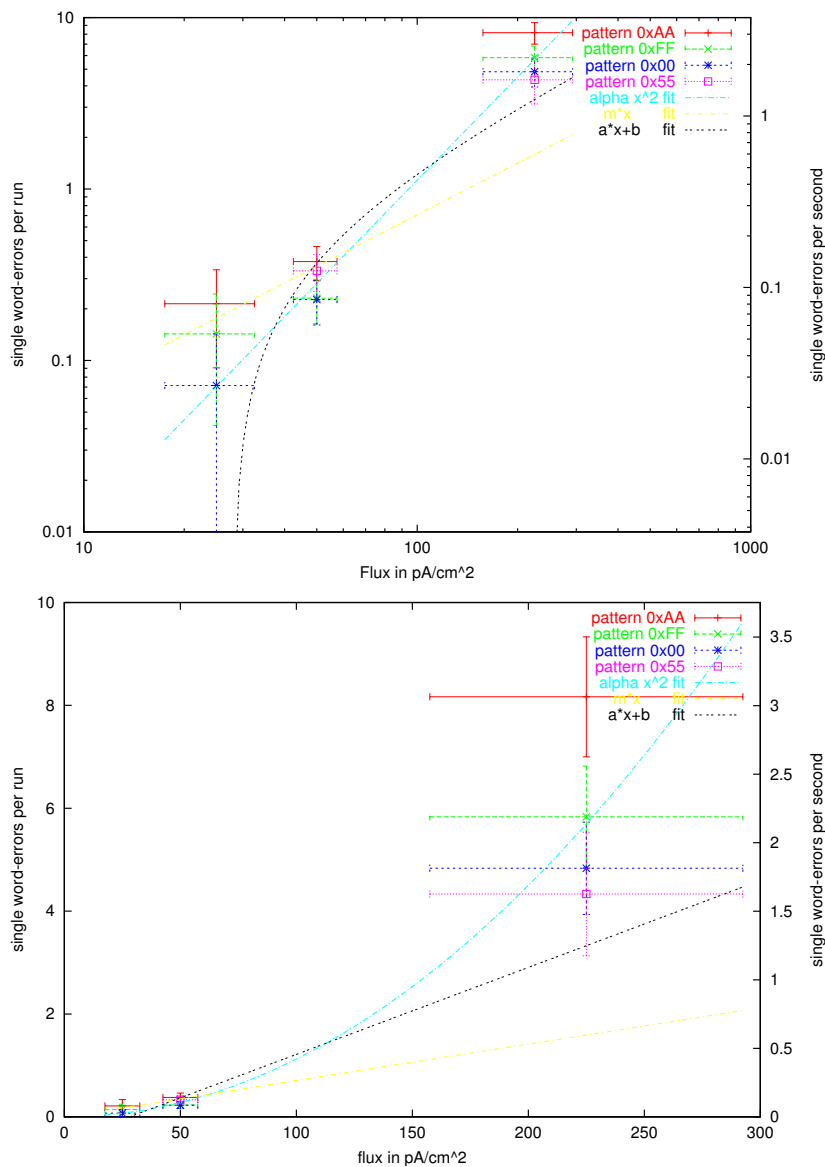


Abbildung 4.15: Flußabhängigkeit der Fehlerrate in SDRAM (Einzelfehler) in doppelt logarithmischer und linearer Darstellung

4.3.2.2 Zweite Testreihe (ohne Betriebssystem)

Die zweite Testreihe ist auf Strommessungen ohne Karte normiert, weswegen die Genauigkeit des Stroms bei selber Schwankungsbreite weit schlechter ist, da eine Rekalibrierung zur Laufzeit somit nur auf einen Wert erfolgen konnte, der an sich schon nur eine entsprechende Genauigkeit hatte. Außerdem ist die Wiederholbarkeit der Stromstär-

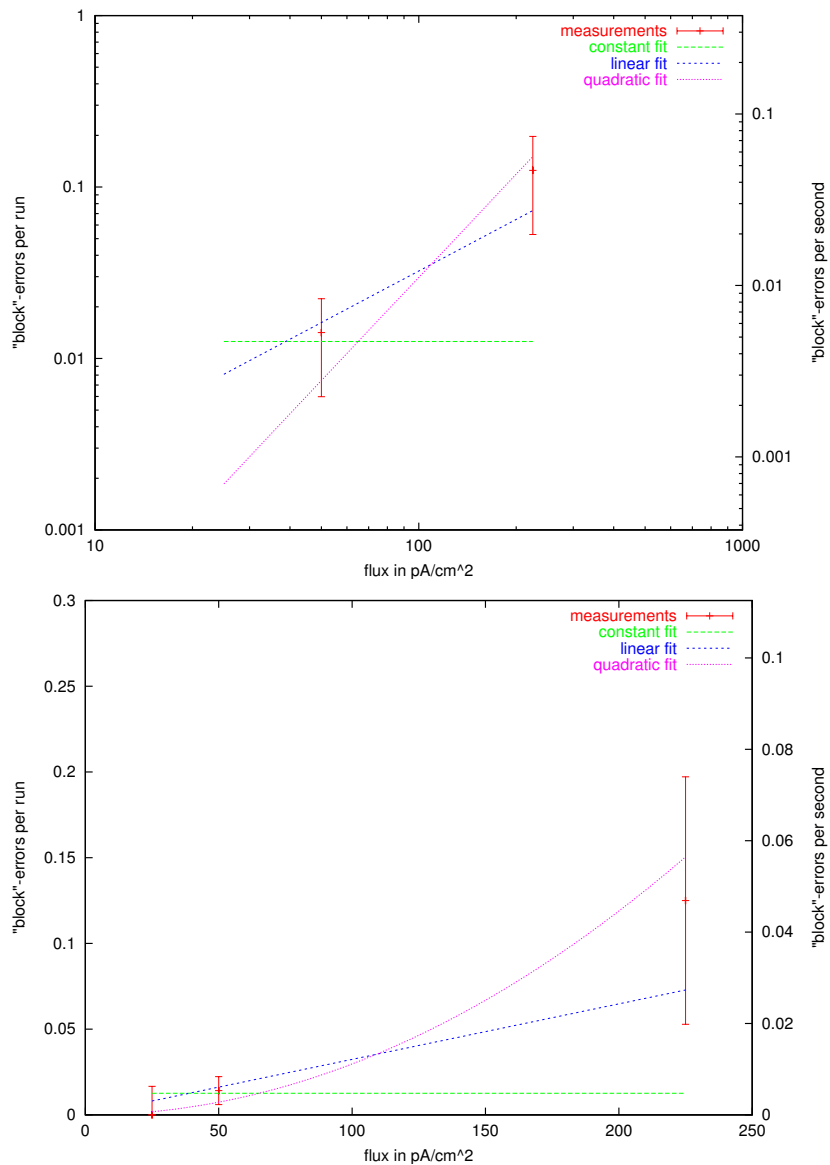


Abbildung 4.16: Flußabhängigkeit der Fehlerrate in SDRAM („Block“-Fehler) in doppelt logarithmischer und linearer Darstellung

ke zwischen verschiedenen Durchläufen nur begrenzt gegeben (siehe 4).
So ist der eingetragene Fehler hier als Genauigkeit des Stroms und nicht als seine Schwankungsbreite zu lesen.

4 Die Messungen und ihre Ergebnisse

Probleme traten vor allem dadurch auf, daß keine gesicherte Datenübertragung vom Testdesign zum speichernden PC erfolgte, da ohne Flußsteuerung gearbeitet wurde. Aus diesem Grunde sind einige Log-Daten verlorengegangen, was zu interessanten Artefakten in den Abbildungen führt (siehe Abbildung 4.17).

Daß es sich hierbei wirklich um Fehler in der Auslese handelt, die nicht direkt aus der

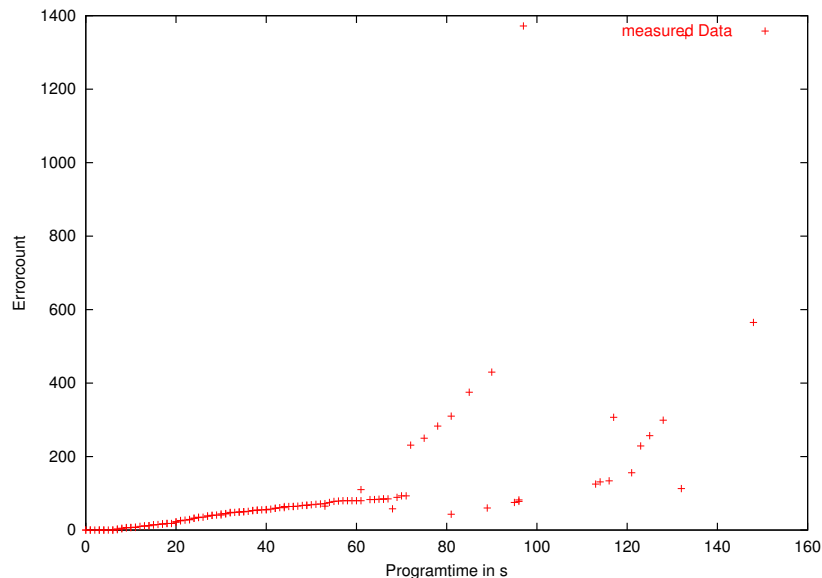


Abbildung 4.17: Fehlerzahl gegen Programm-Laufzeit in SDRAM bei $70 \frac{\mu A}{cm^2}$ (zweite Testreihe) Run1.dat

Bestrahlung resultieren, lässt sich aus den Datenfiles ansehen:

Ausschnitt (ohne Zeitangabe) aus 70pA/run01.dat Zeilen 2977-2979:

```
'ERROR: addr=0x01CAE17C    value=0x01EAE17C'  
'ERROR: addr=0x01CFE6B8    value=0xERROR: addr=0x01DFE4B4    value=0x01F  
'ERROR: addr=0x01E35E58    value=0x01E35E5A'
```

Dies führt zu dem Problem, daß nicht ersichtlich ist wieviele Daten verlorengegangen sind. Deswegen wurden nur die Daten bis eine Sekunde vor einem derartigen Problem verwendet. Die Fits erstrecken sich immer auf den Bereich „Startzeit des Strahls plus eine Sekunde“ bis „Problem- bzw. Endzeit minus eine Sekunde“ (Startzeit Strahl \neq Startzeit Programm!).

Dies liefert dann – nach Bilden der Mittelwerte – Abbildung 4.18.

Achtung: Der Fit wurde hier über die Flußdichtengenauigkeit und nicht über die Fehlerrate vorgenommen! Dies geschah, weil dieser Fehler hier einen weit stärkeren Einfluß

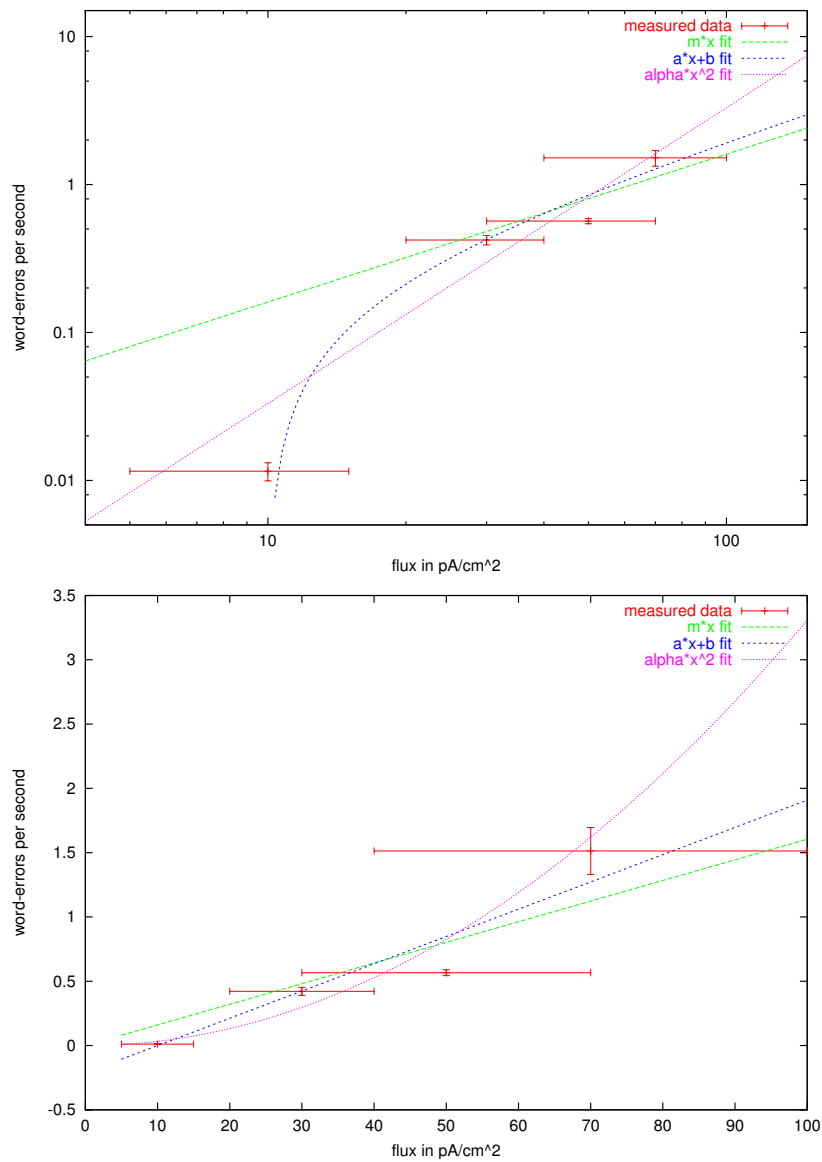


Abbildung 4.18: Flußabhängigkeit der Fehlerrate in SDRAM (zweite Testreihe) in doppelt logarithmischer und linearer Darstellung

auf das Ergebnis ausübt als die relativ genau bekannten Werte der Fehlerraten.

Fitergebnisse Abbildung 4.18

Funktion	Parameter
αx^2	$\alpha = 3.3 * 10^{-4} \frac{\frac{Errors}{s}}{\left(\frac{pA}{cm^2}\right)^2} \pm 26\%$
$m x$	$m = 1.6 * 10^{-2} \frac{\frac{Errors}{s}}{\frac{pA}{cm^2}} \pm 28\%$
$a x + b$	$a = 2.1 * 10^{-2} \frac{\frac{Errors}{s}}{\frac{pA}{cm^2}} \pm 18\%$
	$b = -0.21 \frac{Errors}{s} \pm 39\%$

4.3.2.3 Dritte Testreihe (memtest3 auf DCS-Board)

Da die Strommessung auch hier ohne Karte erfolgte sind auch hier die eingetragenen Fehler in der Flußdichte als Genauigkeit der selbigen zu lesen und nicht als Schwankungsbreite!

Die Plots mit den Fehlerzahlen und den Verteilungen im Adressraum ähneln denen der ersten Testreihe, weswegen ich hier keine weiteren dazu anführe. Lediglich die Flußabhängigkeit der Fehlerrate ist hier interessant. Sie ist in Abbildung 4.19 zu sehen.

Wie man sieht ist auch hier wieder keine klare Entscheidung über linear bzw. quadratisch möglich, da die Karten 2 und 8 zusammen einen eher linearen Verlauf zeigen, während Karte 9 auf einen quadratischen Verlauf hindeutet. Der quadratische Verlauf ist jedoch, da er mit einem Chip gemessen ist weit glaubwürdiger als der lineare, der Messungen mit zwei verschiedene Chips bei verschiedenen Stromdichten bemühen muß.

Die Fits sind trotzdem für *alle* Karten zusammen. In diesem Falle ist der Fit wieder über die Genauigkeit in der Flußdichte vorgenommen worden.

Fitergebnisse Abbildung 4.19

Funktion	Parameter
αx^2	$\alpha = 6.3 * 10^{-5} \frac{\frac{Errors}{s}}{\left(\frac{pA}{cm^2}\right)^2} \pm 47\%$
$m x$	$m = 2.8 * 10^{-3} \frac{\frac{Errors}{s}}{\frac{pA}{cm^2}} \pm 40\%$

Außerdem traten insgesamt vier permanente Fehler auf (Stuck-Bits). Zwei davon heilten jedoch innerhalb von zwei Wochen wieder aus (soweit man davon ausgehen kann, daß das Programm wieder denselben Speicherbereich zugewiesen bekam).

Da die anderen beiden Fehler auch nach ihrem ersten Auftreten weiter bestrahlt wurden lässt sich vermuten, daß diese u.U. noch ausheilen können, bzw. daß sie bei weniger starker Bestrahlung hätten ausheilen können. Aufgrund der geringen Anzahl derartiger Fehler lässt sich keinerlei Aussage über eine Flußabhängigkeit machen.

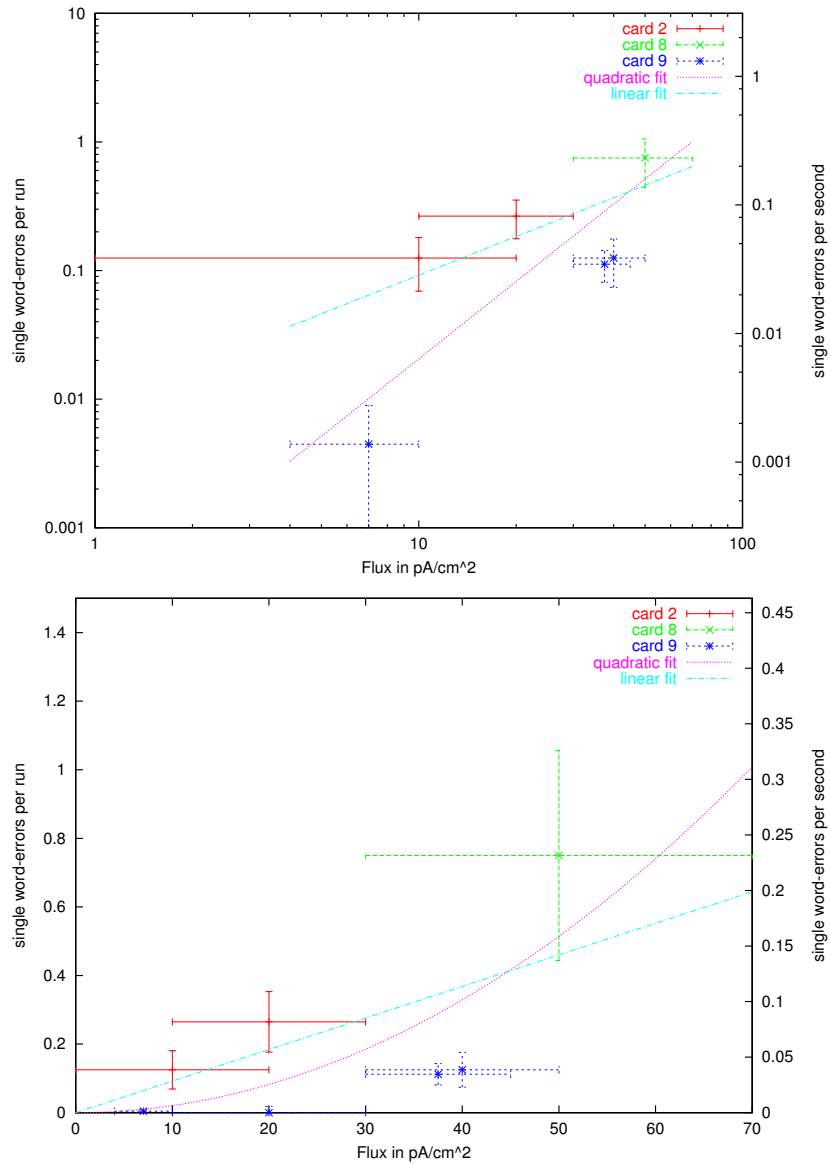


Abbildung 4.19: Flußabhängigkeit der Fehlerrate in SDRAM (DCS-Board) in doppelt logarithmischer und linearer Darstellung

4.4 FLASH

Getestet wurden insgesamt 3 Exemplare folgenden Chips:

- „AM 29LV320 DT -90EI“ auf verschiedenen CIA-RORC-Karten[Ata]

Weiteres siehe Anhang A.3.

4.4.1 Durchführung des Experimentes

Mit dem FLASH wurden insgesamt 3 Tests ausgeführt. Prinzip, Durchführung und Bemerkungen des ersten Tests siehe Kapitel 4.2 (SRAM). Der einzige Unterschied ist, daß die Daten nicht bei jedem Test neu geschrieben wurden, sondern nur einmal vorher.

Der zweite Test unterschied sich vom ersten nur insofern, daß andere Software verwendet wurde und die Flußdichte weit kleiner war.

Die dritte Testreihe diente dazu festzustellen, wie lange das Programmieren eines FLASH-Chips bei Bestrahlung möglich ist. Die Durchführung verlief folgendermaßen:

1. Der „alignment laser“ wurde mit Hilfe der Quarzscheibe ausgerichtet.
2. Die Karte wurde mit dem zu untersuchenden Chip als Ziel montiert.
3. Die Karte wurde programmiert und ihre Funktionsfähigkeit überprüft.
4. Der FLASH wurde gelöscht, neu beschrieben und verifiziert.
5. Der Strahl wurde für wenige Sekunden kurz angeschaltet (inklusive Kalibration auf Nennwert).
6. Der FLASH wurde gelöscht, neu beschrieben und verifiziert.
7. Der Strahl wurde für wenige Sekunden kurz angeschaltet (inklusive Kalibration auf Nennwert).
8. u.s.w.

4.4.2 Die Messergebnisse

4.4.2.1 Erste Testreihe

Nachdem der FLASH mit $10 \frac{\mu A}{cm^2}$ und $100 \frac{\mu A}{cm^2}$ für jeweils etwa 10 Minuten bestrahlt worden war und sich keinerlei Effekt zeigte, wurde er mit $1 \frac{nA}{cm^2}$ bestrahlt. Das Ergebnis ist in Abbildung 4.20 dargestellt.

Der Beginn der Bestrahlung ist ungefähr bei $(80 \pm 20)s$. Interessant ist hier, daß ein

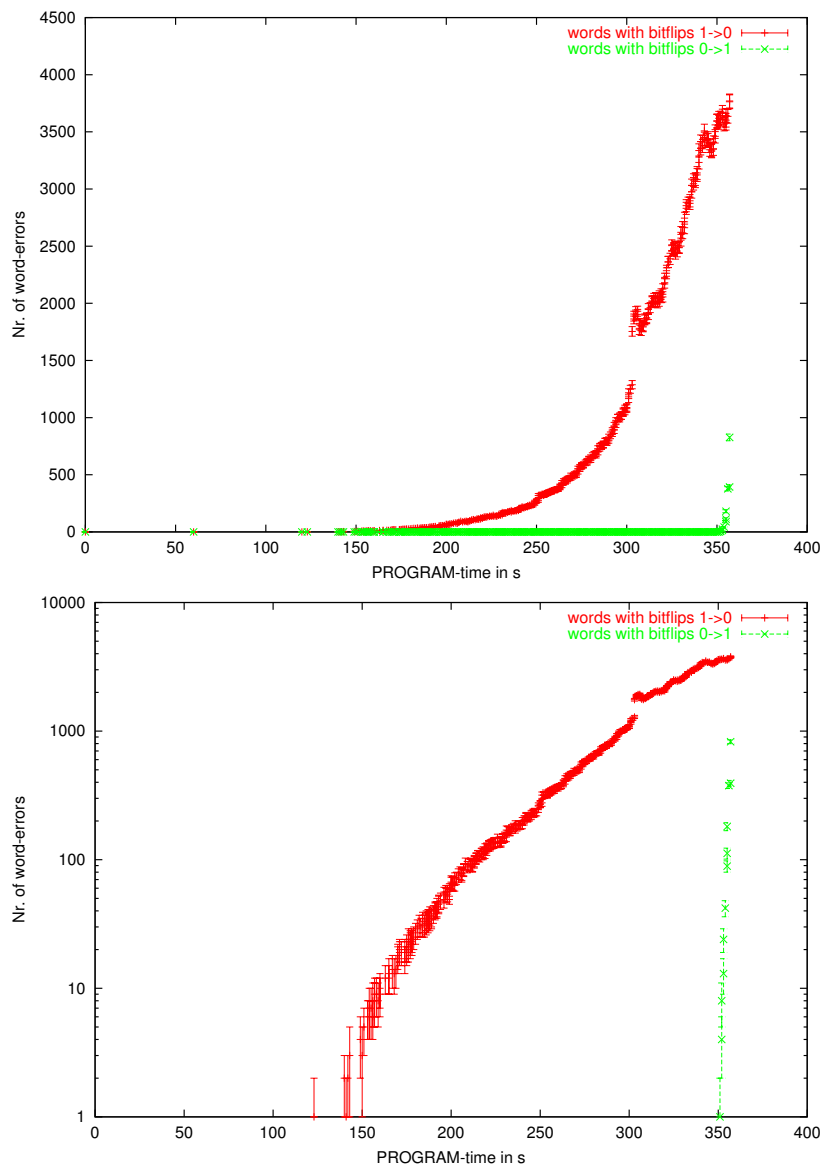


Abbildung 4.20: FLASH bei $1 \frac{nA}{cm^2}$ in einfach logarithmischer und linearer Darstellung

einmal gekipptes Bit nicht zwangsläufig gekippt bleibt. Dies erkennt man daran, daß es hier im Gegensatz zu S(D)RAM auch Rückgänge in der Fehlerzahl gibt (z.B. bei ~310s).

Trägt man die Fehlerposition gegen die Zeit auf, so erhält man Abbildung 4.21. Der Bruch bei Adresse 32k entspricht im übrigen gerade einer Blockgrenze im FLASH. Diese Blöcke werden unabhängig voneinander gelöscht.

Ein größeres Problem jedoch trat auf, als versucht wurde den FLASH nach der Bestrahlung zu löschen und neu zu beschreiben – um zu sehen ob es sich um permanente oder

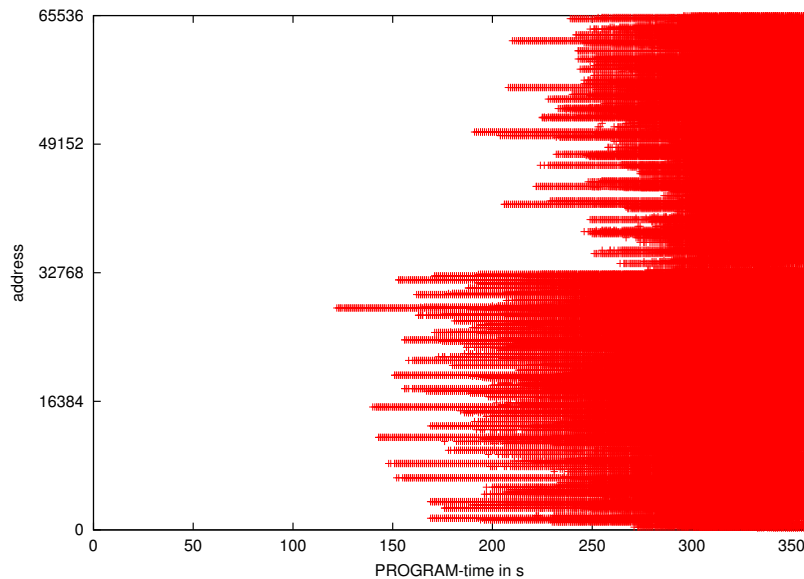


Abbildung 4.21: Fehlerposition gegen Programmlaufzeit im FLASH bei $1 \frac{nA}{cm^2}$

nicht permanente Schäden handelt. Ein Löschen der Daten war unmöglich. Der Chip reagierte auf das Löschkommando, indem er einen konstanten Wert auf die Ausgänge legte und jegliche andere Aktion bis zu einem Power-Cycle verweigerte. Dieser Schaden heilte auch nicht innerhalb von 6 Monaten (Das Verhalten war noch das gleiche). Ein Auslesen war immer noch möglich, bei den Bitflips jedoch waren nach dieser Zeit nur noch 846 $1 \rightarrow 0$ bzw. $2 \ 0 \rightarrow 1$ vorhanden.

4.4.2.2 Zweite Testreihe

Die zweite Testreihe wurde gemacht um zu sehen, ob die entstehenden Bitflips reversibel sind. Zu diesem Zweck wurde der FLASH diesmal mit einer niedrigeren Flußdichte von „nur“ $250 \frac{pA}{cm^2}$ bestrahlt.

Während der Bestrahlung traten zwar einige Lesefehler auf, die Art dieser Fehler jedoch zeigt, daß es sich um Fehler in der Datenübertragung handelt, die aus der nicht vorhandenen Flußsteuerung resultieren.

Beispiel Logfile Zeile 23:

```
Fri Jun 13 18:18:08 2003: Adress 888C: F0F7 -> 0F0F ^= FFF8
```

Jeder dieser Werte ist immer um 4 Bit rechtsgeschiftet, oder anders gesagt, es fehlt immer die letzte Ziffer (Hexadezimal). Nach der Bestrahlung (etwa $(24 \pm 2) Min.$) war

kein echter Fehler aufgetreten, ein Löschen des FLASH war jedoch mit den gleichen Symptomen wie oben unmöglich. Bei Überprüfung der Funktionsweise nach 3 Monaten hatte sich dies nicht geändert.

4.4.2.3 Dritte Testreihe

In der dritten Testreihe ging es nur noch um die Möglichkeit des Löschens und Neuschreibens der Daten. Zu diesem Zweck wurde der FLASH nur jeweils sehr kurz der Strahlung ausgesetzt. Insgesamt wurde bei Intensitäten zwischen $10 \frac{\mu A}{cm^2}$ und $40 \frac{\mu A}{cm^2}$ der FLASH mit etwa $5 \frac{nC}{cm^2} \pm 10\%$ bzw. etwa $3 * 10^{10} \frac{\text{Protonen}}{cm^2}$ bestrahlt. Es gab zwar durchgehend Probleme mit zwei Speicherstellen, diese waren jedoch von Beginn an defekt. Ansonsten lief das Löschen und Programmieren ohne Probleme ab (von offensichtlichen Lesefehlern wie oben einmal abgesehen).

4.5 FPGA

Bei den FPGAs wurden vier verschiedene Typen getestet:

- ein ALTERA ACEX „EP1K100QC208-2“ auf einer ACEX-Karte[Ang01]
- ein ALTERA APEX „EP20K400EFC672-2X“ auf einer CIA-RORC-Karte[Ata]
- ein ALTERA Excalibur ARM „EPXA1F484C1“ auf einem EPXA1-Development-Board[ALTa]
- drei ALTERA Excalibur ARM „EPXA1F484C3“ auf einem DCS-Board

Das FPGA wurde in diesem Test als eine Art Coprozessor verwendet. Siehe hierzu auch Anhang A.4.

Die „utilisation“ war ca. 55% für das APEX und jeweils über 90% für ACEX und EPXA1.

4.5.1 Durchführung des Experimentes

Die Durchführung erfolgte folgendermaßen

1. Der „alignment laser“ wurde mit Hilfe der Quarzscheibe ausgerichtet.
2. Die Karte wurde mit dem zu untersuchenden Chip als Ziel montiert.
3. Die Karte wurde programmiert und ihre Funktionsfähigkeit überprüft.
4. Die Auslese wurde gestartet (Dies impliziert ein Neuprogrammieren des Systems).
5. Der Strahl wurde angeschaltet und auf Nennwert gebracht.
6. Nachdem genügend Fehler aufgetreten waren wurde das FPGA im Strahl neu programmiert und es wurde weitergemessen. (Siehe hierzu auch Anhang A.4 und A.1.
7. Nach genügend vielen Durchläufen wurden Strahl und Programm beendet.

Hierbei waren die Tests mit dem APEX FPGA hauptsächlich zum Testen der prinzipiellen Realisierbarkeit der Strahlentests gedacht, um herauszufinden ob und in wie weit FPGAs durch die Protonenbestrahlung geschädigt werden.

Zum verwendeten Design siehe auch Anhang A.4.

4.5.2 Die Messergebnisse

Trägt man die Fehlerhäufigkeit eines funktionellen Blockes innerhalb des FPGAs gegen seine Größe auf, so erhält man Abbildung 4.22.

Daß sich keine Ursprungsgerade ergibt, sondern der Fit einen Offset zeigt, ist leicht zu

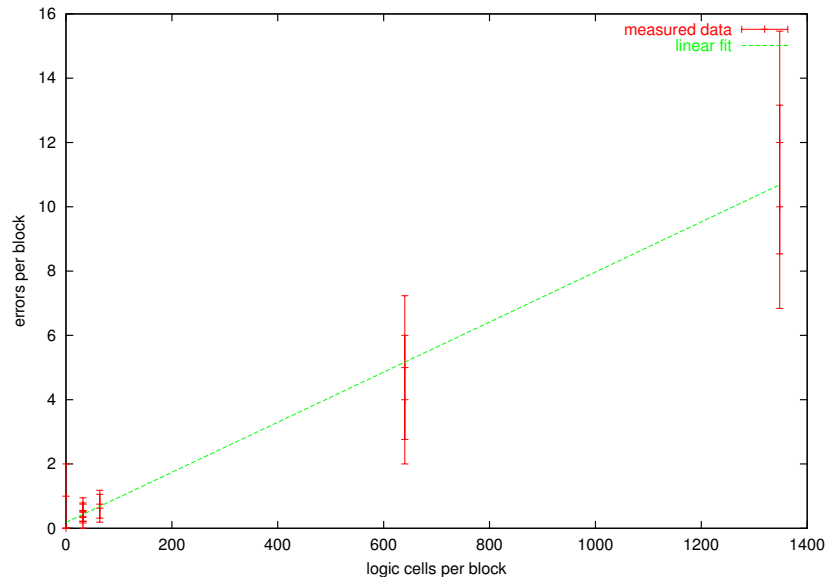


Abbildung 4.22: Fehlerwahrscheinlichkeit gegen Blockgröße im ACEX-FPGA (bei $10 \frac{\mu A}{cm^2}$)

verstehen, wenn man bedenkt, daß das Verbindungsnetzwerk nicht in die Blockgröße eingeht, wohl aber Fehler produzieren kann. Die Fehlerwahrscheinlichkeit ist immer auf die Größe *eines* Blockes normiert, auch wenn *alle* Blöcke eines Typs dafür ausgezählt wurden (so wurden beispielsweise alle Schieberegister gemeinsam ausgezählt und das Ergebnis auf ein Register normiert).

Bitflips in den Datenregistern wurden nicht beobachtet, aber es wurden auch nur 512 Datenbits benutzt – zuzüglich 32-544 Datenbits (je nach Design) im Datenpfad, jedoch maximal zwei 32 Bit-Register in Folge – die jedoch im Gegensatz zu mehreren Kilobytes an Konfiguration stehen.

Die beiden Fehlertypen (Datenfehler bzw. Konfigurationsfehler) lassen sich dadurch unterscheiden, daß ein Datenfehler durch ein überschreiben des Bits behoben werden kann (Was im Falle dieses Designs bei einem der nächsten Durchläufe mit sehr großer Wahrscheinlichkeit geschieht), während ein Konfigurationsfehler nur durch eine Neuprogrammierung behoben werden kann.

4 Die Messungen und ihre Ergebnisse

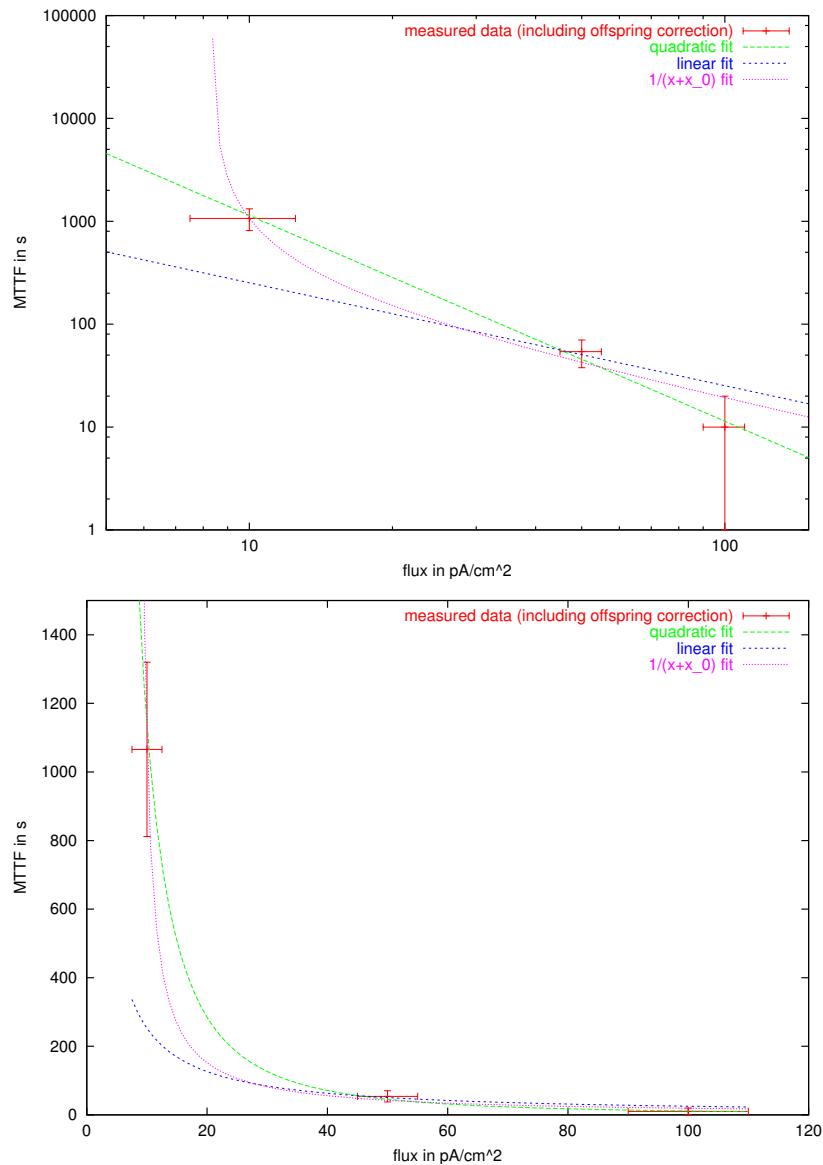


Abbildung 4.23: Flußabhängigkeit der Lebensdauer des Work-Test Designs (APEX) in doppelt logarithmischer und linearer Darstellung

Betrachtet man die Flußabhängigkeit der Lebensdauer des APEX-Designs so ergibt sich Abbildung 4.23.

Auch hier ist der quadratische (diesmal reziprok quadratische) Fit der Beste.

Fitergebnisse Abbildung 4.23

Funktion	Parameter
αx^{-2}	$\alpha = 1.14 * 10^5 s \left(\frac{pA}{cm^2}\right)^2 \pm 8\%$
$m x^{-1}$	$m = 2.5 * 10^3 s \frac{pA}{cm^2} \pm 61\%$
$a(x + x_0)^{-1}$	$a = 1.77 * 10^3 s \frac{pA}{cm^2} \pm 37\%$
	$x_0 = -8.3s \pm 9\%$

Große Diskrepanzen erkennt man, wenn man die mittleren Lebensdauern des ACEX-Designs und des APEX-Designs vergleicht. Hier zeigt sich nämlich ein sehr großer Unterschied, der sich nicht allein durch die Ausnutzung des Bausteins (etwa 50% beim APEX und etwa 100% beim ACEX) erklären läßt. So lebt das ACEX-Design bei $10 \frac{pA}{cm^2}$ im Mittel keine 20s, während das APEX-Design im Mittel über 1000s übersteht. Aus der Ausnutzung des Bausteins aber würde man nur einen Faktor von etwa 2 bis (schätzungsweise) 8 erwarten (wegen der erhöhten Ausnutzung des Verbindungsnetzwerkes steigt die Wahrscheinlichkeit, daß ein Fehler sichtbar ist überproportional an).

Aufgrund der umständlichen Programmierbarkeit des FPGA-Teils im EPXA1 – es mußte zuerst LINUX gebootet werden und dann über einen Web-Client das Design geladen werden – wurden beim ersten Versuch nur sehr wenige Testläufe hiermit ausgeführt. Das Ergebnis dieser Messungen bei $100 \frac{pA}{cm^2}$ war eine MTTF¹ von $8.5s \pm 2.87s$ (statistisch) $\pm 1s$ (systematisch).

Eine systematische Analyse der Flußabhängigkeit konnte mit diesem Verfahren aus Zeitgründen nicht durchgeführt werden.

In einem zweiten Test wurde das gleiche Design auf dem DCS-Board über den FLASH geladen. Die Flußabhängigkeit der MTTF ergibt dann Abbildung 4.24. Achtung: *Die Flußdichte ist hier wieder auf ein Messung ohne Karte normiert!*

Eine eindeutige Entscheidung zwischen linear und quadratisch ist in diesem Falle nicht möglich. Mehr Punkte konnten jedoch nicht sinnvoll gemessen werden, da der systematische Fehler durch die Programmierdauer von 4s schon 2s beträgt. Somit hätte eine Messung höherer Flußdichten keine Erkenntnisse gebracht und eine Messung bei niedrigeren Flußdichten war aufgrund des Beschleunigers nicht möglich.

Die Fits sind wieder über die Abweichung in der Flußdichte erfolgt.

Fitergebnisse Abbildung 4.24

Funktion	Parameter
αx^{-2}	$\alpha = 1.47 * 10^3 s \left(\frac{pA}{cm^2}\right)^2 \pm 16\%$
$m x^{-1}$	$m = 114s \frac{pA}{cm^2} \pm 7\%$

¹„Mean Time To Failure“

4 Die Messungen und ihre Ergebnisse

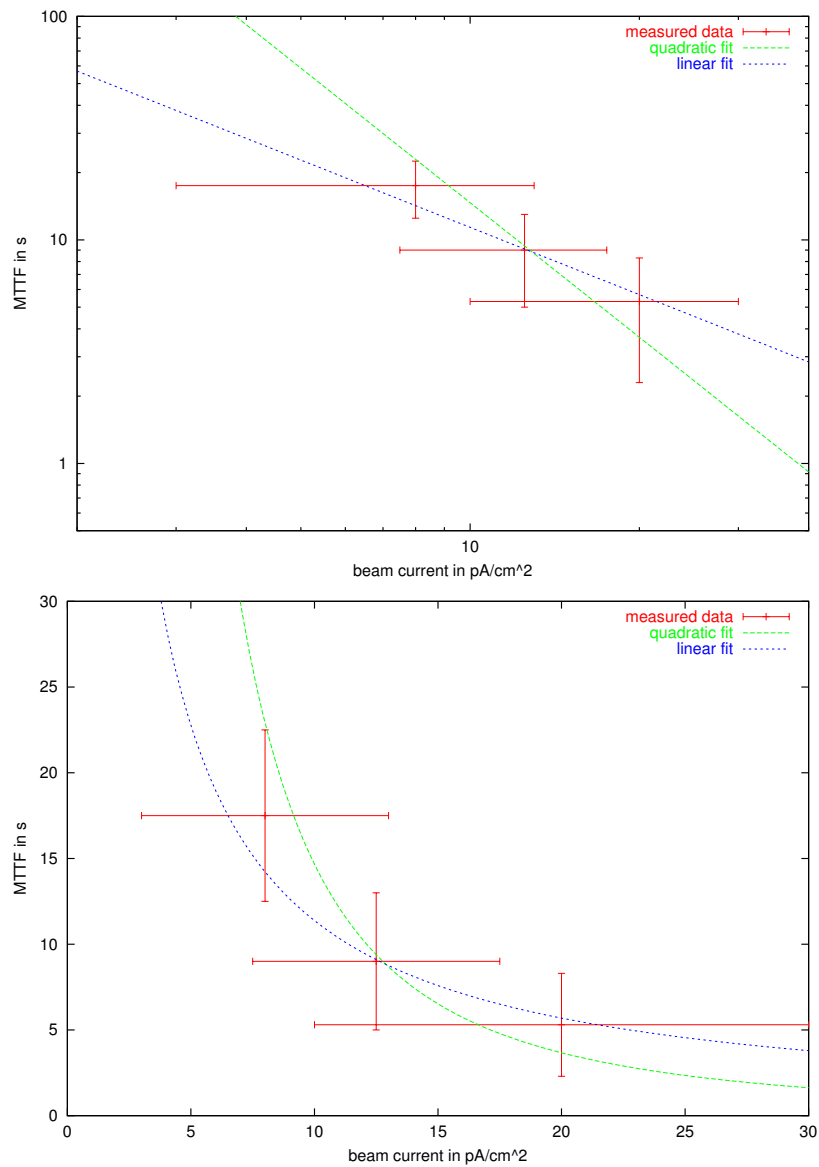


Abbildung 4.24: Flußabhängigkeit der Lebensdauer des Work-Test Designs (EPXA1 auf DCS-Board) in doppelt logarithmischer und linearer Darstellung

4.6 ARM

Es wurde der ARM-Core von folgendem Chip getestet:

- ALTERA Excalibur ARM „EPXA1F484C1“ auf einem EPXA1-Development-Board[ALTa]

4.6.1 Durchführung des Experimentes

Die Durchführung erfolgte folgendermaßen

1. Der „alignment laser“ wurde mit Hilfe der Quarzscheibe ausgerichtet.
2. Die Karte wurde mit dem zu untersuchenden Chip als Ziel montiert.
3. Die Karte wurde (aus dem FLASH) programmiert und ihre Funktionsfähigkeit überprüft.
4. Die Auslese wurde gestartet
5. Der Strahl wurde angeschaltet und auf Nennwert gebracht.
6. Wenn Fehler auftraten wurde der Strahl ausgeschaltet.
7. Die Karte wurde (aus dem FLASH) programmiert und ihre Funktionsfähigkeit überprüft.
8. Die Auslese wurde gestartet
9. u.s.w.

4.6.2 Die Messergebnisse

Aus den Messergebnissen ergibt sich die Abbildung 4.25. Allerdings wurden als Fehler nur solche Änderungen im Programmablauf gewertet, die deutlich ersichtlich waren, also etwa ein Stopp des Programms oder das Einstellen der Funktion.

Der „schönste“ Fehler war jedoch folgender:

```
Fri Jun 13 11:27:44 2003:'00010000'  
Fri Jun 13 11:27:44 2003:'%08X'
```

4 Die Messungen und ihre Ergebnisse

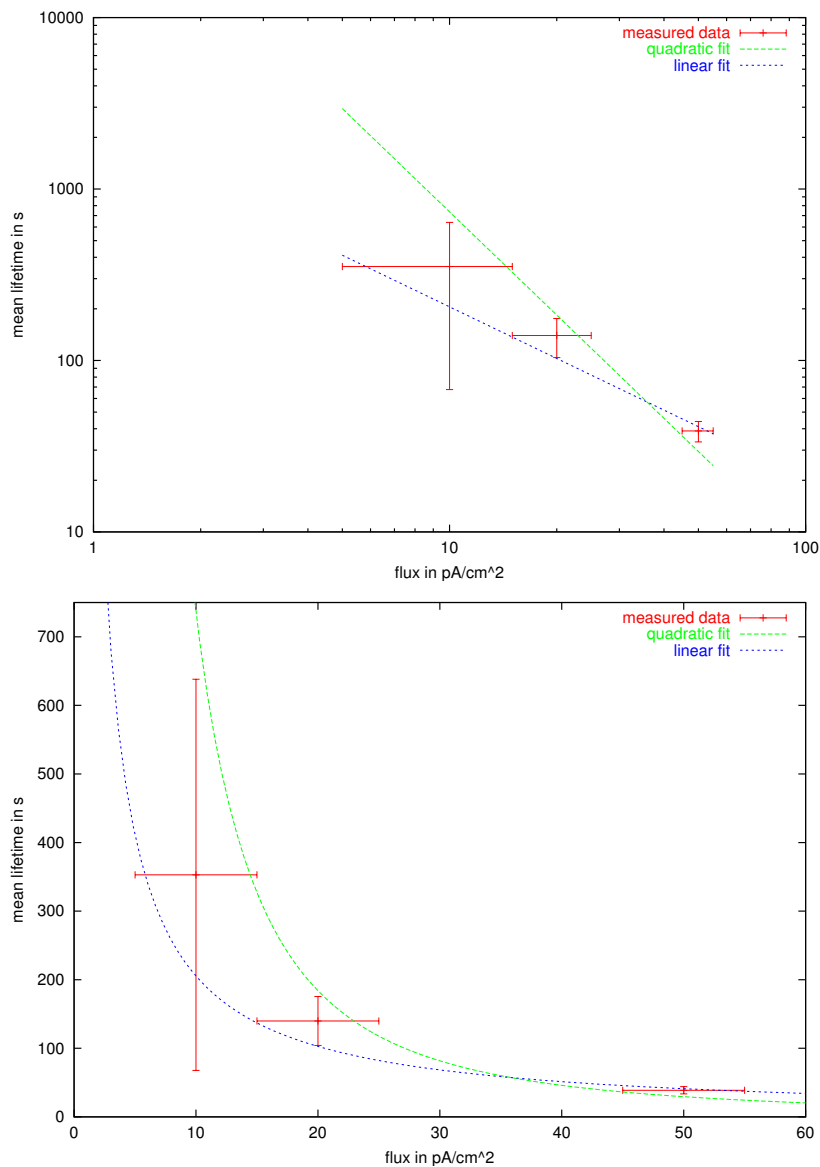


Abbildung 4.25: Flußabhängigkeit der ARM-Lebensdauer in doppelt logarithmischer und linearer Darstellung

(Zu finden in 50pA_run9.dat) Das Programm gibt also den „Formatstring“ der 'printf' Anweisung unverändert aus und wertet die Format-Anweisung nicht aus, wie es sollte.

Fitresultate Abbildung 4.25

Funktion	Parameter
αx^{-2}	$\alpha = 7.4 * 10^4 \text{ s} \left(\frac{\text{pA}}{\text{cm}^2}\right)^2 \pm 22\%$
$m x^{-1}$	$m = 2.05 * 10^3 \text{ s} \frac{\text{pA}}{\text{cm}^2} \pm 11\%$

Zu beachten ist hier noch, daß es aufgrund der Architektur des EPXA1 nicht möglich war den ARM-Core unabhängig von seinem FPGA-Teil zu untersuchen. Zwar ist es prinzipiell möglich den ARM ohne ein geladenes Design zu betreiben (Das FPGA kann auch vom ARM aus programmiert werden), jedoch ist eine Kommunikation mit dem ARM dann nur sehr schwer möglich. Auch können ARM und FPGA nicht getrennt von einander bestrahlt werden, da sie sich auf dem selben Stück Silizium befinden.

4.7 TRAP

Es wurden die interne Speicher verschiedener Exemplare folgenden Chips getestet.

- TRAP1 auf Test MCM¹s.

4.7.1 Durchführung des Experimentes

Es wurden zwei Strahlentests mit TRAP-Chips durchgeführt. Der erste Test wurde von mir durchgeführt, der zweite von Dr. Heinz Tilsner. In beiden Fällen wurde Software von Dr. Venelin Angelov (KIP) für die Tests verwendet. Die Auslese beim ersten Test erfolgte mit der von Robin Gareus entworfenen Hard- und Software[Gar02]. Beim zweiten Test wurde auch für die Auslese von Dr. Venelin Angelov entwickelte Hard- und Software verwendet. Die Durchführung erfolgte beim ersten Test wie folgt:

1. Der „alignment laser“ wurde mit Hilfe der Quarzscheibe ausgerichtet.
2. Die Karte wurde mit dem zu untersuchenden Chip als Ziel montiert.
3. Die Karte wurde programmiert und ihre Funktionsfähigkeit überprüft.
4. Der Strahl wurde eingeschaltet und auf Nennwert gebracht.
5. Nach entsprechender Zeit wurde die erste Messung beendet und ein anderes Test-Programm auf den Chip geladen.
6. u.s.w.

4.7.2 Die Messergebnisse

4.7.2.1 Erste Testreihe

Die erste Testreihe wurde bei $10 \frac{\text{pA}}{\text{cm}^2}$ durchgeführt und ergab – vom Nichtauftreten permanenter Schädigungen abgesehen – keine sinnvoll verwertbaren Erkenntnisse. Dies liegt daran, daß der Chip sich unter Strahleneinfluß nicht nachvollziehbar entgegen seiner Programmierung verhielt. Bis zu einem Powercycle war hierbei das Verhalten *wiederholbar*. Im Falle der Speichertests schien der Chip die Daten beim Einschalten des Strahls gelöscht zu haben, was jedoch erst bei der Analyse der Daten auffiel. Beim Einschalten war der Strahlstrom jedoch höher als $10 \frac{\text{pA}}{\text{cm}^2}$.

Ein Vergleich des Programms im Speicher mit dem Originalprogramm zeigte im Falle der Softwaretests allerdings keine Unterschiede, was das Fehlverhalten um so rätselhafter

¹„Multi Chip Module“

macht.

Insgesamt wurde der Chip etwas über eine Stunde bestrahlt, ohne daß permanente Schädigungen auftraten.

4.7.2.2 Zweite Testreihe

Zum Thema Messunsicherheiten und Normierung siehe Abschnitt 4.3.2.2.

Für eine sinnvolle Fehlerratenabschätzung traten nur im Event-Buffer genügend Fehler auf. Diese sind allerdings nur bei langsamer Taktung funktionsfähig. Daher wurde nur Experimente mit langsamer Taktung verwendet. Die Fehlerzahl in den einzelnen Messungen ist jedoch auch hier sehr gering (vor allem bei $10 \frac{pA}{cm^2}$), weswegen die Auswertungen der einzelnen Messreihen nicht über Geraden-Fits erfolgt ist sondern lediglich durch

$\frac{\text{Fehlerzahl}}{\text{Laufzeit}}$.
Zwar wäre bei den höheren Flußdichten und Laufzeiten auch ersteres möglich gewesen, dadurch wäre jedoch die Datenbasis sehr zusammengeschrumpft und eine Mischung beider Verfahren macht in gemeinsamen Plots – aufgrund verschiedener systematischer Fehlerquellen – wenig Sinn. Bei der Bildung des Mittelwertes wurde mit dem Kehrwert des theoretisch zu erwartenden Meßfehlers ($\Delta = \frac{\sqrt{\text{Fehlerzahl}}}{\text{Laufzeit}}$) gewichtet. Alles in allem ergibt dies Abbildung 4.26.

Zwar ist in diesem Falle der $ax + b$ Fit die beste Approximation an die Daten, es ist jedoch fraglich, ob die Fehlerrate unterhalb von $\sim 8.64 \frac{pA}{cm^2}$ wirklich auf Null fällt. Aus diesem Grund denke ich, daß auch hier der quadratische Fit am besten ist.

Fitergebnisse Abbildung 4.26

Funktion	Parameter
αx^2	$\alpha = 7.3 * 10^{-5} \frac{\frac{Errors}{s}}{\left(\frac{pA}{cm^2}\right)^2} \pm 15\%$
mx	$m = 1.5 * 10^{-3} \frac{\frac{Errors}{s}}{\frac{pA}{cm^2}} \pm 69\%$
$ax + b$	$a = 5.1 * 10^{-3} \frac{\frac{Errors}{s}}{\frac{pA}{cm^2}} \pm 6\%$
	$b = -0.044 \frac{Errors}{s} \pm 8\%$

Betrachtet man nicht die Fehlerraten sondern die MTTF, so ergibt sich Abbildung 4.27. Zwar ist hier für die beiden höheren Stromdichten der lineare Verlauf die bessere Approximation, aber die Messwerte widersprechen auch nicht dem quadratischen Verlauf. Daß auch hier der quadratische Verlauf die wahrscheinlichere Interpretation ist, ergibt sich vor allem dann, wenn man bedenkt, daß die systematische Ungenauigkeit in der Zeit vermutlich nicht symmetrisch ist, sondern die Abweichung zu höheren Lebensdauern deutlich kleiner ist als in Richtung niedrigerer Lebensdauern. Außerdem passen die Werte der Vorfaktoren der quadratischen Fits sehr gut zusammen, was für die linearen nicht gilt.

4 Die Messungen und ihre Ergebnisse

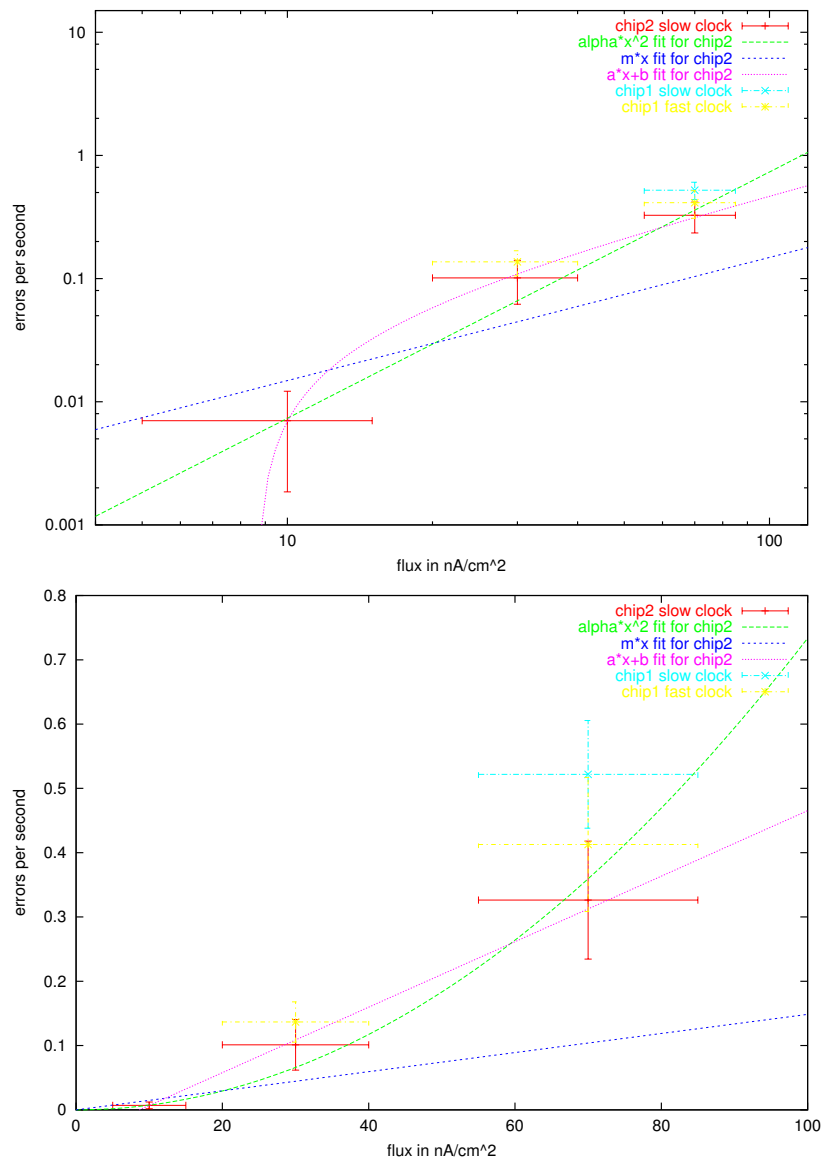


Abbildung 4.26: Flußabhängigkeit der Fehlerrate in TRAP1 Eventbuffern in doppelt logarithmischer und linearer Darstellung

Fitresultate Abbildung 4.27

Funktion	Parameter
αx^{-2}	$\alpha = 1.2 * 10^4 s \left(\frac{\mu A}{cm^2}\right)^2 \pm 2\%$
$a x^{-1}$	$a = 900 s \frac{\mu A}{cm^2} \pm 31\%$

Vergleicht man die Ergebnisse der Fitparameter von Fehlerrate (Abbildung 4.26) und MTTF (Abbildung 4.27), so sieht man, daß im Gegensatz zu den Werten des linearen

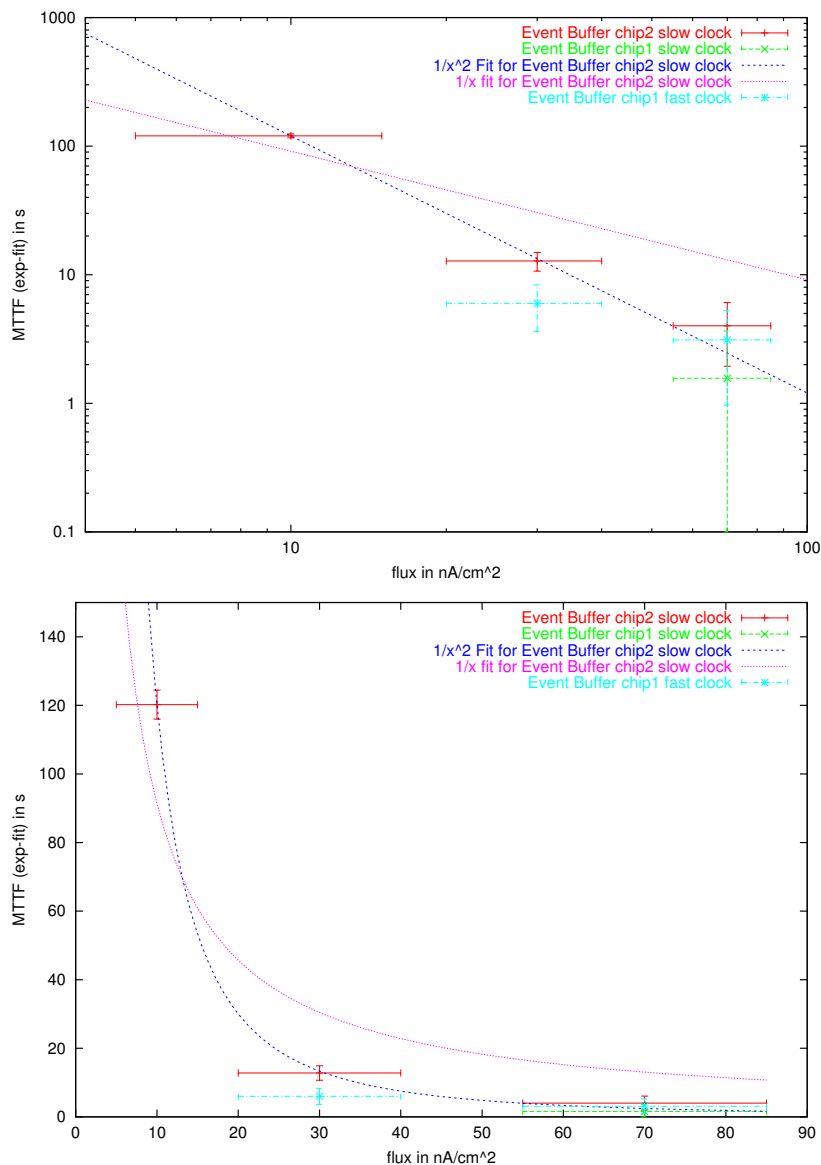


Abbildung 4.27: Flußabhängigkeit der MTTF in TRAP1 Eventbuffern in doppelt logarithmischer und linearer Darstellung

Verlaufs (mx bzw. mx^{-1}) die Werte für den quadratischen Verlauf (αx^2 bzw. αx^{-2}) innerhalb der Fehlergrenzen konsistent sind ($\alpha_{Fehlerrate} = \alpha_{MTTF}^{-1}$).

Achtung: Die Fits beziehen sich alle auf Chip 2 mit niedriger Taktung, da von Chip 1 zu wenig Daten vorhanden sind.

Abgesehen von den Eventbuffern traten zwar noch weitere Datenfehler auf, ihre Zahl war

aber zu gering um Dinge wie Flußabhängigkeit oder ähnliches daraus abzuleiten. Die MTTF der Lookup-Tables war bei der höchsten Flußdichte von $70 \frac{\mu A}{cm^2}$ etwa 20-25 (linear) bzw. 33-55 (nichtlinear) mal so groß wie für die Eventbuffer. Im Instruktions- und Datenspeicher traten etwa doppelt so viele Lesefehler wie Bitflips auf. Es ist unbekannt, ob der Fehler in der Datenübertragung (SCSN¹[Gar02]) oder im Speicher selber liegt. Insgesamt traten in den 20 Testläufen des Chips 2 bei $70 \frac{\mu A}{cm^2}$ 6 Bitflips und 11 Lesefehler auf. Die Gesamtlebensdauer der TRAP-Speicher wird komplett von den Eventbuffern dominiert. Eine Betrachtung zur Flußabhängigkeit der anderen Speicher ist jedoch nicht möglich.

Die MTTF der Lookup-Tables sind bei $(70 \pm 15) \frac{\mu A}{cm^2}$ etwa $(60 \pm 20)s$ (linear) bzw. $(110 \pm 35)s$ (nichtlinear).

Was nicht nachvollziehbares Verhalten anbelangt, so tritt dieses auch bei der zweiten Testreihe auf, allerdings nur bei den höheren Flußdichten und auch dort zu selten, als daß eine sinnvolle quantitative Aussage darüber gemacht werden könnte. Permanente Schädigungen wurden nicht beobachtet.

¹„Slow Control Serial Network“

5 Die Ergebnisse der Flußabhängigkeit Fits

Hier noch einmal die Ergebnisse der Flußabhängigkeit Fits Zusammengefasst.

Chip	Abbildung	Funktion	Parameter
Fehlerrate MT-SRAM 2^{20} Bit	4.7	αx^2	$\alpha = 1.122 * 10^{-4} \frac{\frac{Errors}{s}}{(\frac{pA}{cm^2})^2} \pm 1.1\%$
		ax	$a = 8.0 * 10^{-3} \frac{\frac{Errors}{s}}{\frac{pA}{cm^2}} \pm 25\%$
		$mx + b$	$m = 1.17 * 10^{-2} \frac{\frac{Errors}{s}}{\frac{pA}{cm^2}} \pm 9\%$
			$b = -0.11 \frac{Errors}{s} \pm 18\%$
Fehlerrate SDRAM 2^{26} Bit $^{+10\%}_{-50\%}$ 2^{27} Bit $^{+0\%}_{-50\%}$ 2^{27} Bit $^{+0\%}_{-50\%}$	4.15	αx^2	$\alpha = 4.2 * 10^{-5} \frac{\frac{Errors}{s}}{(\frac{pA}{cm^2})^2} \pm 8\%$
		mx	$m = 2.7 * 10^{-3} \frac{\frac{Errors}{s}}{\frac{pA}{cm^2}} \pm 26\%$
		$ax + b$	$a = 6.3 * 10^{-3} \frac{\frac{Errors}{s}}{\frac{pA}{cm^2}} \pm 26\%$
			$b = -0.18 \frac{Errors}{s} \pm 42\%$
	4.18	αx^2	$\alpha = 3.3 * 10^{-4} \frac{\frac{Errors}{s}}{(\frac{pA}{cm^2})^2} \pm 26\%$
		mx	$m = 1.6 * 10^{-2} \frac{\frac{Errors}{s}}{\frac{pA}{cm^2}} \pm 28\%$
		$ax + b$	$a = 2.1 * 10^{-2} \frac{\frac{Errors}{s}}{\frac{pA}{cm^2}} \pm 18\%$
			$b = -0.21 \frac{Errors}{s} \pm 39\%$
4.19	αx^2	$\alpha = 6.3 * 10^{-5} \frac{\frac{Errors}{s}}{(\frac{pA}{cm^2})^2} \pm 47\%$	
	mx	$m = 2.8 * 10^{-3} \frac{\frac{Errors}{s}}{\frac{pA}{cm^2}} \pm 40\%$	
Blockfehlerrate SDRAM 2^{26} Bit $^{+10\%}_{-50\%}$	4.16	αx^2	$\alpha = 3.0 * 10^{-6} \frac{\frac{Errors}{s}}{(\frac{pA}{cm^2})^2} \pm 28\%$
		mx	$m = 3.2 * 10^{-4} \frac{\frac{Errors}{s}}{\frac{pA}{cm^2}} \pm 28\%$

5 Die Ergebnisse der Flußabhängigkeit Fits

Chip	Abbildung	Funktion	Parameter
MTTF APEX	4.23	αx^{-2}	$\alpha = 1.14 * 10^5 s \left(\frac{pA}{cm^2}\right)^2 \pm 8\%$
		$m x^{-1}$	$m = 2.5 * 10^3 s \frac{pA}{cm^2} \pm 61\%$
		$a(x + x_0)^{-1}$	$a = 1.77 * 10^3 s \frac{pA}{cm^2} \pm 37\%$
			$x_0 = -8.3 s \pm 9\%$
MTTF EPXA1	4.24	αx^{-2}	$\alpha = 1.47 * 10^3 s \left(\frac{pA}{cm^2}\right)^2 \pm 16\%$
		$m x^{-1}$	$m = 114 s \frac{pA}{cm^2} \pm 7\%$
MTTF ARM Core	4.25	αx^{-2}	$\alpha = 7.4 * 10^4 s \left(\frac{pA}{cm^2}\right)^2 \pm 22\%$
		$m x^{-1}$	$m = 2.05 * 10^3 s \frac{pA}{cm^2} \pm 11\%$
TRAP Eventbuffer	4.26	αx^2	$\alpha = 7.3 * 10^{-5} \frac{Errors}{s} \left(\frac{pA}{cm^2}\right)^2 \pm 15\%$
		$m x$	$m = 1.5 * 10^{-3} \frac{Errors}{\frac{pA}{cm^2}} \pm 69\%$
		$a x + b$	$a = 5.1 * 10^{-3} \frac{Errors}{\frac{pA}{cm^2}} \pm 6\%$
			$b = -0.044 \frac{Errors}{s} \pm 8\%$
	4.27	αx^{-2}	$\alpha = 1.2 * 10^4 s \left(\frac{pA}{cm^2}\right)^2 \pm 2\%$
		$a x^{-1}$	$a = 900 s \frac{pA}{cm^2} \pm 31\%$

6 Schlußfolgerungen

6.1 Allgemeines

In den meisten Fällen ist die beste Approximation an die Meßwerte eine quadratische Abhängigkeit von der Flußdichte, und es gibt keine Meßreihen, die einem quadratischen Verlauf widersprechen. Desweiteren sind im Gegensatz zu den linearen Varianten die quadratischen konsistent, wenn sowohl MTTF als auch Fehlerrate bestimmt wurden.

Die kanonische Interpretation einer quadratischen Abhängigkeit ist, daß es sich beim dominierenden Prozess nicht um Einzeltreffer handelt sondern um Doppeltreffer. Die Ladungsdeposition von einem Proton bei 28.5 MeV scheint also nicht zu reichen um einen Fehler auszulösen, oder der Wirkungsquerschnitt hierfür ist bei den verwendeten Flußdichten weit kleiner als der für Doppeltreffer. Die berechneten linearen Wirkungsquerschnitte deuten jedoch darauf hin, daß dies eigentlich unmöglich ist, da schon die gemessenen kleiner sind als vorhandene Strukturen.

Ein großes Problem der Wirkungsquerschnitte der quadratischen Prozesse ist jedoch die beteiligte Zeitkonstante, innerhalb derer dieser Doppeltreffer erfolgen muß. Diese muß aus anderer Quelle (z.B. Simulation einzelner Speicherzellen oder den Datenblättern) bezogen werden.

6.2 Berechnung der Wirkungsquerschnitte

6.2.1 Wirkungsquerschnitt für lineare Prozesse

Der Wirkungsquerschnitt für einen linearen Prozeß lässt sich aus folgender Beziehung berechnen:

$$N_H = N_P \times \frac{N_T \times \sigma}{A}$$

Wobei:

6 Schlußfolgerungen

N_H : Anzahl der Treffer

N_P : Anzahl der eingestrahnten Teilchen

N_T : Anzahl der Ziele

σ : Wirkungsquerschnitt eines Ziels

A : Größe der bestrahlten Fläche

Man geht dabei von einer Gleichverteilung der eingestrahnten Teilchen über A aus, und daß die Anzahl der Ziele konstant bleibt oder $N_T \gg N_H$.

Dies führt durch Auflösen und Einsetzen von

$$N_P = F \times A \times T$$

mit:

F : Flußdichte (Teilchen je Fläche und Zeit)

T : Strahlzeit

zu folgendem Ergebnis:

$$\sigma = \frac{N_H}{N_T \times F \times T}$$

6.2.2 Wirkungsquerschnitt für quadratische Prozesse

Bei einem Prozess mit Doppeltreffer ist die Berechnung eines Wirkungsquerschnittes etwas komplizierter. Die Ausgangsgleichung hier lautet wie folgt (Anzahl der Einzeltreffer mal Wahrscheinlichkeit innerhalb von ΔT einen zweiten Treffer zu erzielen):

$$N_H = N_P \times \frac{N_T \times \sigma_1}{A} \times \frac{\sigma_2}{A} \times F \times A \times \Delta T$$

Mit:

σ_1 : Wirkungsquerschnitt für den ersten Treffer

σ_2 : Wirkungsquerschnitt für den zweiten Treffer

ΔT : Maximale Zeitdifferenz zwischen den Ereignissen

Sonstige s.o.

Hieraus lässt sich nun das Produkt der beiden Wirkungsquerschnitte berechnen:

$$\sigma_1 \sigma_2 = \frac{N_H}{N_T \times F^2 \times T \times \Delta T}$$

Mit Hilfe dieser Formeln und der Bedingung einfach geladener Teilchen – was für Protonen ja der Fall ist – lassen sich nun aus den Fitergebnissen Wirkungsquerschnitte berechnen. (MTTF $\approx \frac{1}{\text{Fehlerrate}}$)

$$F = \frac{I}{A} \frac{1}{e_c}$$

$\frac{I}{A}$: Gemessene Stromdichte

e_c : Elektronenladung $1.602 \times 10^{-19} C$

6.3 Bestimmung der Zeitkonstanten für Doppeltreffer für SDRAM

Die Zeit innerhalb derer der Doppeltreffer erfolgen muß ist in erster Näherung durch die Refresh-Zyklen gegeben. Die mittlere zur Verfügung stehende Zeit ist die Hälfte der Refresh Zeit, welche für den verwendeten SDRAM $64ms$ beträgt. Es ist allerdings nicht auszuschließen, daß ein signifikanter Teil der Fehler nicht im DRAM passiert, sondern in den angeschlossenen Latches für die Zeilenpufferung. In diesem Falle wäre die Fehlerrate weniger von der Größe des benutzten Speichers abhängig, als von der Anzahl der bestrahlten Bänke (Genauer: Anzahl der bestrahlten Latches). Außerdem wäre die Zeitkonstante vermutlich mit der für den SRAM vergleichbar.

6.4 Bestimmung der Zeitkonstanten für Doppeltreffer für SRAM

Um eine Zeitkonstante zu erhalten innerhalb derer der Doppeltreffer erfolgen muß, wurde eine TRAP Eventbuffer SRAM Zelle simuliert (Schaltplan siehe Abbildung 6.1). Als Modell für die durch die ionisierende Strahlung eingebrachten Ladungen wurden Stromquellen verwendet, die das Einbringen von Ladung in Form des Kurzschließen des Transistors mit Masse (bzw. V_{dd}) durch eine Ladungsdeposition simulieren sollen (in diesem Falle mit Masse).

6 Schlußfolgerungen

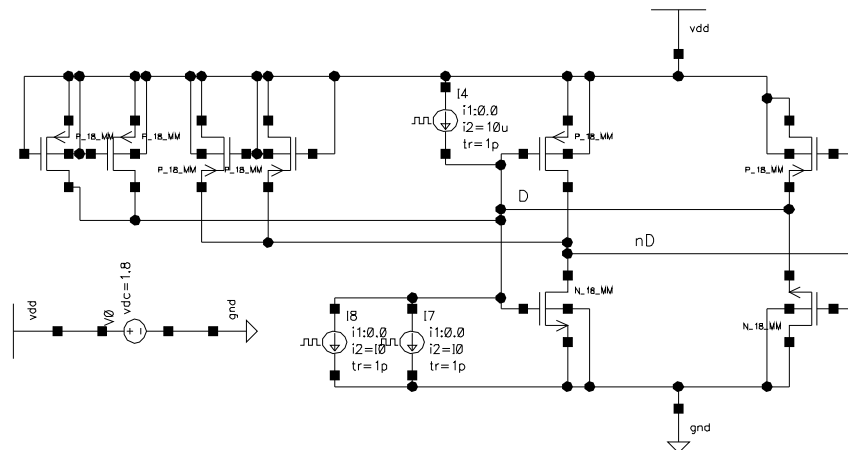


Abbildung 6.1: Schaltplan für die Simulation des SRAM Bitflips im TRAP Eventbuffer mit Stromquellen als Strahlungersatz. Rechts die gegengekoppelten Inverter, links die Auslesetransistoren (Ein Lese-, Ein Schreibport).

Über den p-MOS Transistor sind hierfür (bei 10 ps Pulsen) etwa 1,545 mA nötig, über den n-MOS etwa 1,17 mA. Dies entspricht etwa 50.000 bzw. 35.000 Elektron-Loch Paaren. Implantiert man – über den n-MOS – zweimal 10ps lang jeweils 1mA (etwa 30.000 Elektron-Loch Paare), so ergibt sich eine maximale Zeitdifferenz zwischen den Pulsen von etwa 300ps, damit die Zelle umkippt (siehe Abbildung 6.2).

6.5 Gemessene Wirkungsquerschnitte

In der folgenden Tabelle sind die nach den obigen Formeln aus der Flußabhängigkeit berechneten Wirkungsquerschnitte zusammengefasst.

Chip	Abbildung	Funktion	Wirkungsquerschnitt σ bzw. $\sqrt{\sigma_1\sigma_2}$
MT-SRAM	4.7	αx^2	$10 \frac{\mu m^2}{Bit}$ $\Delta T = 300ps$
		ax	$0.13 \frac{\mu m^2}{2^{20} Bit}$ $\pm 25\%$
		$mx + b$	$0.187 \frac{\mu m^2}{2^{20} Bit}$ $\pm 9\%$
SDRAM	4.15	αx^2	$0.6 \frac{\mu m^2}{2^{26} Bit^{+10\%}_{-50\%}}$ $\Delta T = 32ms$
			$47 \frac{\mu m^2}{Latch^{+10\%}_{-50\%}}$ $\Delta T = 300ps$
		mx	$0.043 \frac{\mu m^2}{2^{26} Bit^{+10\%}_{-50\%}}$ $\pm 26\%$
		$ax + b$	$0.10 \frac{\mu m^2}{2^{26} Bit^{+10\%}_{-50\%}}$ $\pm 26\%$

6.5 Gemessene Wirkungsquerschnitte

Chip	Abbildung	Funktion	Wirkungsquerschnitt σ bzw. $\sqrt{\sigma_1\sigma_2}$	
	4.18	αx^2	$1.6 \frac{\mu m^2}{2^{27} \text{Bit}^{+0\%}}$	$32ms$
			$93 \frac{\mu m^2}{\text{Latch}^{+0\%}_{-50\%}}$	$\Delta T = 300ps$
		mx	$0.26 \frac{\mu m^2}{2^{27} \text{Bit}^{+0\%}}$	$\pm 28\%$
		$ax + b$	$0.34 \frac{\mu m^2}{2^{27} \text{Bit}^{+0\%}}$	$\pm 18\%$
	4.19	αx^2	$0.7 \frac{\mu m^2}{2^{27} \text{Bit}^{+0\%}}$	$32ms$
			$40 \frac{\mu m^2}{\text{Latch}^{+0\%}_{-50\%}}$	$\Delta T = 300ps$
		mx	$0.045 \frac{\mu m^2}{2^{27} \text{Bit}^{+0\%}}$	$\pm 40\%$
	SDRAM Blockf.	4.16	αx^2	$1600 \mu m^2$
$13 \frac{\mu m^2}{\text{Latch}^{+10\%}_{-50\%}}$				$\Delta T = 300ps$
mx			$0.0051 \frac{\mu m^2}{2^{26} \text{Bit}^{+10\%}}$	$\pm 28\%$
APEX 20K400E	4.23	αx^{-2}	$2700 \mu m^2$	$\Delta T = 300ps$
		mx^{-1}	$0.0064 \mu m^2$	$\pm 61\%$
		$a(x + x_0)^{-1}$	$0.0091 \mu m^2$	$\pm 37\%$
MTTF EPXA1	4.24	αx^{-2}	$24,000 \mu m^2$	$\Delta T = 300ps$
		mx^{-1}	$0.14 \mu m^2$	$\pm 7\%$
MTTF ARM Core	4.25	αx^{-2}	$2,700 \mu m^2$	$\Delta T = 300ps$
		mx^{-1}	$0.0078 \mu m^2$	$\pm 11\%$
TRAP Eventbuffer	4.26	αx^2	$7.900 \mu m^2$	$\Delta T = 300ps$
		mx	$0.024 \mu m^2$	$\pm 69\%$
		$ax + b$	$0.082 \mu m^2$	$\pm 6\%$
	4.27	αx^{-2}	$8400 \mu m^2$	$\Delta T = 300ps$
		ax^{-1}	$0.018 \mu m^2$	$\pm 31\%$

Für die quadratischen Querschnitte wird hier kein Fehler angegeben, da die Genauigkeit des ΔT direkt in den Gesamtfehler eingeht, die benutzten $300ps$ aber auch für die TRAP Eventbuffer – den einzigen Speicher für den sie motiviert sind – nur eine sehr grobe Abschätzung sind. Ein Fehler um einen Faktor 2 ist wahrscheinlich, eine Größenordnung ohne weiteres möglich.

Ein weiteres Indiz dafür, daß eher die quadratischen Verläufe die Realität widerspiegeln ist, daß die Wirkungsquerschnitte für die linearen Prozesse sehr klein sind, während die der quadratischen Prozesse eher zu Strukturen auf den Chips passen können. So ist bei einem $0,13 \mu m$ Prozess die kleinste erzeugbare Größe (Fläche des Gates eines minimalen

6 Schlußfolgerungen

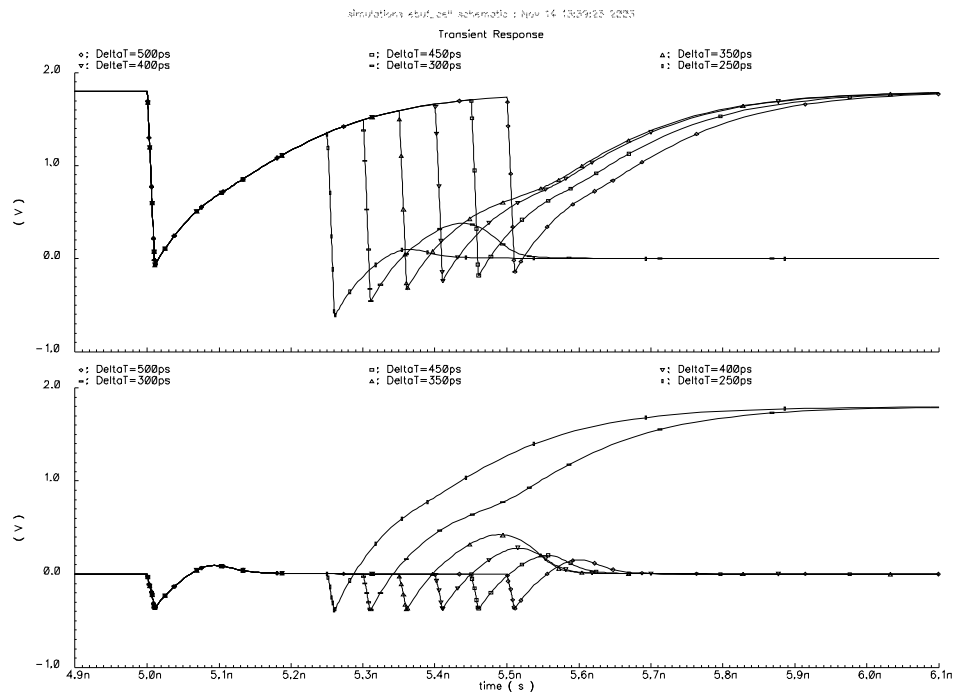


Abbildung 6.2: Simulation des SRAM Bitflips im TRAP Eventbuffer

Transistors) größer als $(0,13\mu\text{m})^2 = 0,017\mu\text{m}^2$. Auch der geometrische Querschnitt der Sperrschicht ist selbst für einen minimalen Transistor weit größer (Dicke der Sperrschicht ist in der Größenordnung (Schätzung) von $0,04\mu\text{m}$). Die berechneten Querschnitte würden also darauf hindeuten, daß auch bei Benutzung von 16MB SDRAM nur die Gates von weniger als 5 Transistoren als mögliche Schadensquellen dienen. Die Querschnitte der quadratischen Prozesse lassen sich weit besser mit echten Strukturen in Verbindung bringen.

Zwar haben auch die quadratischen Querschnitte nicht immer passende Größen, aber die Größenordnungen passen weit besser zu vorhandenen Strukturen. So ist für den TRAP Eventbuffer zwar die Größe der Zelle mit etwa $16\mu\text{m}^2$ noch deutlich kleiner als der mittlere Wirkungsquerschnitt von etwa $68\mu\text{m}^2$ je Bit, liegt damit aber dennoch weit besser als $1,8\text{nm}^2$ je Bit für den linearen Prozess.

Der Wert für den MT-SRAM ist mit einem berechneten Wirkungsquerschnitt (quadratischer Prozeß) von etwa $10\mu m^2$ sogar sehr nahe an dem was man erwarten würde (unter der Voraussetzung, daß die Bits eine vergleichbare Größe wie im TRAP haben). Der systematische Faktor in der Stromstärke von etwa 2 – der auf die unterschiedliche Normierung der Messungen zurückzuführen ist – ist hier auch nicht weiter störend, da die Zeitkonstante weit weniger genau ist.

Betrachtet man den SDRAM so sieht man, daß die Querschnitte der Bits wieder weit kleiner sind als echte Strukturen. Dies legt nahe, daß die Fehler unter Umständen eher bei den gelatchten Daten passieren, also beim Refresh oder während der Auslese. Die Zeitkonstante für die „Block“-Fehler ist hier nie mit 32ms angenommen, da aufgrund der Korrelation zwischen den Adressen angenommen werden kann, daß dieser Fehler nicht in den einzelnen Bits sondern irgendwo in den Spalten auftritt.

Die meisten Messungen deuten auf ein quadratisches Verhalten hin, und es gibt keine Messung, welche einem quadratischen Verhalten widerspricht. Wenn dies ein systematischer Meßfehler sein sollte, so müßte er das quadratische Verhalten haben. Auch wenn aufgrund der Anzahl der Punkte argumentiert werden könnte, daß der Nullpunkt einen Offset besitzt, so ist auch dies nicht wahrscheinlich, da die Schwankungen teilweise auch unter diesen (errechneten) Wert fielen und die Anzeige bei Abschalten des Strahls auf Null zurückging. Alles in allem bleibt zu sagen, daß keine der an den Messungen beteiligten Personen ein entsprechendes Modell für einen derartigen Fehler hat.

6.6 Abschließende Bemerkungen

Abschließend kann man sagen, daß bei den zu erwartenden Flußdichten von effektiv etwa $7 * 10^9 \frac{\text{Protonen}}{\text{cm}^2}$ in 10 Jahren (Strahlzeit Pb-Pb 10 Monate bei 8 kHz Eventrate)[Sto] recht wenige Fehler auftreten werden. Allerdings gibt es ein paar einschränkende Kriterien. Es wurden lediglich die Effekte von monoenergetischen Protonen untersucht. Dies führt dazu, daß Protonen mit leicht geringerer Energie u.U. einzeln ausreichen um Bit-Flips auszulösen. Dies könnte sehr große Probleme verursachen wenn die quadratische Hypothese zutrifft. Allerdings handelt es sich auch schon bei den verwendeten Protonen fast um „Maximum Ionizing Particles“. d.h. sie befinden sich fast auf dem Maximum des Ionisations-Energieverlusts. Es ist also nicht sicher, daß derartiges passieren kann. Desweiteren wurden die Chips nicht alle über die ganze Fläche bestrahlt (möglicherweise nicht nur der SDRAM sondern auch die anderen Chips bei denen es keine auffällige Abweichungen gibt). Dies führt dazu, daß die Lebensdauern sich möglicherweise deutlich verkürzen. Es steht nicht zu erwarten, daß es Änderungen um eine Größenordnung geben wird.

6 Schlußfolgerungen

Permanente Schädigungen sind – zumindest in der getesteten Hardware – nicht in signifikantem Maße zu erwarten. Es kann nicht ausgeschlossen werden, daß einzelne permanente Bitfehler im SDRAM auftreten. Die entsprechenden Chips waren bei Auftreten der Fehler schon mit deutlich mehr als $10^{10} \frac{\text{Protonen}}{\text{cm}^2}$ bestrahlt worden. Es kann auch nicht ausgeschlossen werden, daß dieser Effekt flußabhängig ist, also erst bei höheren Flußdichten auftritt (Eine Potenz deutlich größer als zwei in der Flußabhängigkeit ist möglich!). Außerdem ist ein Großteil der vorhandenen Defekte wieder ausgeheilt, was im Falle von ALICE möglicherweise dazu führt, daß diese Art von Fehlern gar nicht erst auftritt, da die Analogeffekte ausheilen, bevor sie digital sichtbar werden.

Es kann auch nicht ausgeschlossen werden, daß einzelne FLASHs zum Ende der Laufzeit nicht mehr gelöscht werden können. Damit sollte man aber leben können. Möglicherweise tritt dieses Problem aber auch nur bei höheren Flußdichten (Größenordnung $100 \frac{\text{pA}}{\text{cm}^2}$) auf und ist somit für das Experiment irrelevant. Der TRAP scheint – gerade auch wegen seiner Hammingcodierung auf den Instruktionen und Datenspeichern – strahlungsunkritisch zu sein. Zwar steht es zu erwarten, daß in den Eventbuffern „häufiger“ Bitfehler auftreten (Worstcase Abschätzung, lineares Verhalten):

$$\text{MTTF}_{\text{TRAP}} \approx 6 \text{ Monate}$$

Aufgrund der hohen Anzahl der Chips ist allerdings ein Fehler etwa alle 4 Minuten zu erwarten (gesamter Detektor). Diese Fehler werden jedoch in den seltensten Fällen sichtbar sein, da relevante Daten dort sehr bald ausgelesen werden. Außerdem ist ein Fehler in den Eventbuffern in den meisten Fällen nicht sichtbar, da der Test über den kompletten Speicher lief (64 Timebins * 21 Kanäle * 10 Bit).

Bitflips in Instruktions- und Datenspeicher können leicht dadurch repariert werden, daß die Chips selbständig die Daten auslesen und zurückschreiben. Zumindest Einzelbitfehler im Datenspeicher lassen sich hierdurch trivial korrigiert. Für den Instruktionsspeicher lassen sich ähnliche Programme schreiben, die – wenn häufig genug ausgeführt – die Wahrscheinlichkeit für Abstürze minimieren dürften.

Bei den DCS-Boards steht zu erwarten, daß sie innerhalb der 10 Jahre mehrfach ausfallen. Nimmt man hier ein exponentielles Zerfallsgesetz der Einzelkomponenten an (Was zumindest für die FPGA-Designs nicht zu gelten scheint und für den FLASH falsch ist), und sagt, daß jeglicher Fehler kritisch ist (Was im SDRAM nicht der Fall ist) so läßt sich eine effektive Lebensdauer des Board berechnen mit:

$$e^{-\frac{t}{\tau_{ges}}} = \prod_i e^{-\frac{t}{\tau_i}} \quad \Leftrightarrow \quad \tau_{ges}^{-1} = \sum_i \tau_i^{-1}$$

Mit diese Formel und der Näherung $\text{MTTF} = \frac{1}{\text{Fehlerrate}}$ für den SDRAM erhält man dann aus den Werten von ARM-Core, FPGA, FLASH und SDRAM eine MTTF für das DCS-Board. Nicht berücksichtigt wird hierbei, daß es auch in den gemachten Messungen Wechselwirkungen zwischen ARM und FPGA Teil gibt.

Unter der Worst-Case Abschätzung (lineares Verhalten) folgt:

$$\text{MTTF}_{\text{DCS-Board}} \approx 21 \text{ Tage}$$

Dies übersetzt sich dann in einen Fehler je 40 Minuten für den gesamten Detektor. Allerdings ist es – wie oben angesprochen – sehr wahrscheinlich, daß sich die Lebensdauern nicht linear mit der Flußdichte verändern. Wenn dies so ist, so verlängert sich die Lebensdauer um einen Faktor von etwa $2 * 10^6$.

Weitere Tests mit Komponenten des DCS-Board und TRAP2 wurden von Dr. Marc Stockmeier (PI¹), Dirk Gottschalk (KIP) und Phillip Struck durchgeführt bzw. sind in Planung.

¹„Physikalisches Institut“Universität Heidelberg

Anhang

A Designs & Software

Die FPGA-Designs wurden mit Quartus (Altera) angefertigt und übersetzt. Die Designs und die Software sind in den entsprechenden Verzeichnissen auf der beigelegten DVD zu finden.

A.1 Kontroll Script

Das Script 'control' ist für die automatische Ausführung der Memory-Test Software entworfen und dann für die Work-Test Programme erweitert worden. Es protokolliert die Start und Endzeiten der Testprogramme und startet die Software im Falle eines unplanmäßigen Abbruchs automatisch neu. Es ist in Perl geschrieben und erwartet vier Argumente:

```
'control PROGRAMM OPT1 DATEINAME OPT3'
```

Hierbei ist:

PROGRAMM ist der auszuführende Befehl. Er wird direkt an eine Shell zum Ausführen übergeben.

OPT1 wird direkt als erstes Argument für den Befehl in PROGRAMM verwendet und einfach weitergereicht.

DATEINAME wird als Dateiname interpretiert (UNIX/LINUX Form). Er wird verwendet, um die Namen der Logdatei zu bestimmen (DATEINAME.log), und wird in erweiterter Form als zweites Argument des Befehls in PROGRAMM verwendet. Zu diesem Zweck wird zwischen Dateiname und letzter Endung eine weitere Endung eingefügt, die von Null aufwärts fortlaufend durchzählt. Ein Dateiname wird nur dann verwendet, wenn eine Datei selben Namens noch nicht existiert.

OPT3 wird wie OPT1 direkt als drittes Argument an den in PROGRAMM enthaltenen Befehl weitergereicht.

Zum Thema Perl und reguläre Ausdrücke siehe z.B. [tJO] und [Fri].

A.2 SCSN-Treiber

Viele der Testprogramme benutzen als Kommunikationsweg zum Computer das von Robin Gareus entwickelte SCSN[Gar02]. Jedoch war die von ihm entworfene Software für dieses Einsatzgebiet nicht geeignet, da auch falsche und fehlende Pakete ohne lange Wartezeiten behandelbar sein mußten. Aus diesem Grunde wurde auf Basis des von Robin Gareus programmierten EXCALIBUR-SIMULATOR[Gar02] von mir ein eigener PCI-SCSN Adapter auf einer CIA-RORC[Ata] entwickelt.

Mit diesem kann über PCI direkt auf die Anschlüsse des EXCALIBUR-SIMULATOR[Gar02] zugegriffen werden.

A.2.1 Das Design

Die CIA-RORC[Ata] Karte verwendet PCI und benutzt hierfür die normale Altera ID: (Ausschnitt aus 'scsn.h')

```
#define VENDOR_ID    0x1172
#define DEVICE_ID    0x0004
```

Der Adressraum der Karte erstreckt sich über 1kB.

Das Design besteht – von etwas Dekodier/Kontroll Logik und der Blink-Anzeige abgesehen – praktisch nur aus dem Altera PCI-Core pci_mt64[ALTe] und dem EXCALIBUR-SIMULATOR[Gar02]. Der 64Bit PCI-Core wurde verwendet, da für einen 32Bit PCI-Core keine Lizenz vorhanden war. Dies macht keine größeren Probleme. Das Design ist allerdings für 32Bit PCI entwickelt und es ist unbekannt, ob es in einem 64Bit Slot funktionsfähig ist (auch unabhängig von der dort verwendeten Taktfrequenz von 66 MHz). Die Adressierung des EXCALIBUR-SIMULATORs[Gar02] erfolgt über die Bits Fünf bis Zwei der PCI-Adresse. Die Bits Neun bis Sechs werden auf Null getestet, die Bits Eins und Null ignoriert. Bei Zugriff auf die Adresse 256 im internen Adressraum wird ein Reset für den EXALIBUR-SIMULATOR[Gar02] ausgelöst.

Später wurde für die Tests mit dem DCS-Board noch die Funktion hinzugefügt, bei einem SCSN reset auch ein ca. 1ms langes reset Signal über den Disketten-Anschluß der Karte (Pins 7 & 8 ?!) auszugeben um das DCS-Board aus der Ferne neu starten zu können.

A.2.2 Das Softwareinterface

Das Softwareinterface ist in C geschrieben und basiert auf den PSI¹-Treibern[Eng]. Es erlaubt einen direkten Zugriff auf Paketebene. Zu den Funktionen siehe 'scsn.h'. Zu Testzwecken sind einige dieser Funktionen auch mit dem Programm 'scsn' auf der Kommandozeile verfügbar.

A.3 Memory-Test Design(s)

Das Memory-Test Design wurde in drei Varianten benutzt, die sich nur im angeschlossenen Speicher, bzw. dem FPGA und der Karte unterscheiden. Alle Designs arbeiten mit derselben Software und wurden via SCSN[Gar02] und einer Controller-Karte (Siehe Abschnitt A.2) angeschlossen.

Die drei Varianten waren:

mtest_ACEX_SRAM : Zum Testen des auf der ACEX-Karte[Ang01] montierten IDT-SRAMs.

mtest_APEX_SRAM : Zum Testen des auf der CIA-RORC[Ata] montierten MT-SRAMs.

mtest_APEX_FLASH : Zum Testen des auf der CIA-RORC[Ata] montierten FLASH.

A.3.1 Das 'mtest' FPGA Design

Das FPGA-Design besteht logisch aus 4 Blöcken:

- Die Daten Generierung 'dgen'
- Der Speicher mit Controller
- Die Datenanalyse 'analyse'
- Das SCSN-Slave-Design 'scsn_slave'

Hierzu kommen noch einige Register zum Synchronisieren der verschiedenen Laufzeiten sowie die Taktgenerierung und ein paar Anpassungen der Busbreiten.

¹„PCI & Shared memory Interface“

A.3.1.1 'dgen.vhd'

Das 'dgen'-Design generiert mit Hilfe des 'hash'-Designs (Siehe A.3.1.5) pro Takt ein Adresse-Daten Paar. Die Adresse wird hierbei linear hochgezählt. Die Bitbreiten von Adresse und Daten lassen sich mittels der Generics 'Adr_width' und 'Dat_width' einstellen. Außerdem wird die Art des Hashing durch den Parameter 'Hash_typ' definiert (siehe A.3.1.5).

Für einen kompletten Adressdurchlauf (nach jedem Reset) wird außerdem ein Write-Enable Signal 'WE' gesetzt. Nach Ende eines Durchlaufs (im ersten Takt eines neuen Durchlaufs) wird außerdem für einen Takt ein Neustart-Signal 'restart' zum Synchronisieren der Komponenten generiert. Außerdem wird das synchrone Reset Signal 'rst' lauffzeitkorrekt durchgereicht.

A.3.1.2 Der Speicher

Der Speicher wird analog zu einem 'altdpram' verwendet. Hierbei wird davon ausgegangen, daß pro Takt ein Datum verarbeitet werden kann. Dies wurde für die externen SRAM-Chips dadurch erreicht, daß der RAM¹-Controller eine höhere Taktung erhielt.

A.3.1.3 'analyse.vhd'

Das 'analyse'-Design ist für die Zusammenarbeit mit dem 'dgen'-Design (Siehe A.3.1.1) konzipiert. Es geht davon aus, daß alle Signale denselben taktsynchronen Laufzeiten unterliegen und zählt mit Hilfe von 'hash' (Siehe A.3.1.5) die Anzahl der Wortfehler – getrennt nach Richtung des Bit-Flips – aus und gibt ihre Position aus. Einmal je Adressdurchlauf – bei gesetztem 'restart' Signal – werden die internen Zähler nach außen übergeben. Wie das 'dgen'-Design sind die Bitbreiten und der Hash-Typ über Generics konfigurierbar.

A.3.1.4 'scsn_slave'

Das 'scsn_slave'-Design stellt die Kommunikationsschnittstelle für das Memory-Test Design zur Verfügung. Zu diesem Zweck wird dem SCSN-core mcm_network_interface[Gar02] ein Adressmultiplexer und ein FIFO² vorgeschaltet. In dieses FIFO werden die Adressen

¹„Read Access Memory“

²„First In First Out“

und Bitmasken der gefundenen Fehler eingefügt (16 Bit Adresse und 16 Bit Bitmaske in einem 32 Bit Wort). Außerdem dient es als Grenze zwischen den verschiedenen Taktzonen. Der Ausgang des FIFO ist über Adresse Null im Adressraum des Gerätes erreichbar. Alle anderen Adressen sind mit den Fehlercounter Ausgängen des 'analyse'-Designs verbunden, deren unterste 16Bits jeweils in den Bits 31-16 bzw. 15-0 zu finden sind. Ist jeweils das 16. Bit gesetzt (Das Design ist diesbezüglich starr auf 16 Adressbits ausgelegt), so wird -1 statt des entsprechenden Fehlercounters gelesen.

A.3.1.5 'hash.vhd'

Das 'hash'-Design implementiert vier verschiedene Hashfunktionen, die über das Generic 'hash_typ' ausgewählt werden können. Die Bitbreiten der Ein- und Ausgangsbusse können über die Generics 'win' bzw. 'wout' bestimmt werden.

Die Hashfunktionen im einzelnen:

hash_typ=1 :

Diese Hashfunktion ist so entworfen worden, daß auch bei weniger Ausgangs- als Eingangsbits zwei Hashwerte sich in mindestens einem Bit unterscheiden, wenn die Eingangswerte sich um ein Bit unterscheiden.

Dies wurde getan, da am Anfang der Untersuchungen die Anzahl der Adress- und Datenbits nicht feststand und auch Fehler erkannt werden sollten, bei denen beim Auslesen ein Adressbit im Eingang falsch interpretiert wurde.

Zu diesem Zweck wird jedem Bit der Hashfunktion die XOR Verknüpfung aller Bits des Eingangs an gleicher oder höherer Position zugewiesen:

$$\forall 0 \leq i \leq w_{out} - 1 : hash_i(x_{(w_{in}-1)..0}) = \bigoplus_{j=i}^{w_{in}-1} x_j$$

hash_typ=2 :

Für Testzwecke:

$$\forall 0 \leq i \leq w_{out} - 1 : hash_i(x_{(w_{in}-1)..0}) = 0$$

hash_typ=3 :

Für Testzwecke:

$$\forall 0 \leq i \leq w_{out} - 1 : hash_i(x_{(w_{in}-1)..0}) = 1$$

sonst :

Direktes mappen der Augabebits auf die Eingabe. Hierbei werden überzählige Eingabebits ignoriert und überzählige Ausgabebits auf 0 gesetzt.

$$\forall 0 \leq i \leq \min(w_{out}, w_{in}) - 1 : hash_i(x_{(w_{in}-1)..0}) = x_i$$

$$\forall w_{in} \leq i \leq w_{out} - 1 : hash_i(x_{(w_{in}-1)..0}) = 0$$

A.3.2 'mtest.c'

Die Memory-Test Software baut auf meinem SCSN-Interface (Siehe A.2) auf. Sie benutzt die ncurses Bibliothek um Tastatureingaben zu verarbeiten.

Parameter siehe Quellcode oder Fehlerausgabe.

Das Programm ruft den JamPlayer[?] auf, um das als erstes Argument übergebene Design über Port 0x378 (erster Parallel-Port) zu programmieren. Schlägt der Versuch mit entsprechendem Rückgabewert mehrfach fehl, so wird das Programm mit Fehler beendet. Ansonsten sendet das Programm Lesebefehle über SCSN[Gar02] für die Adressen Null und Eins (Siehe hierzu auch A.3.1).

Hierbei wird solange von Adresse Null gelesen, bis das Datum 0x00000000 zurückgeliefert wird, höchstens jedoch $4 * count_warn$, wobei `count_warn` der dritte Parameter des Programms ist. Alle Daten werden hierbei im Logfile (zweiter Parameter des Programms) gespeichert. Im ersten Falle (0x00000000 gelesen) wird dann von Adresse Eins die Fehlerzahl gelesen und – wenn sie von dem zuletzt gelesenen Wert abweicht oder wenn mindestens 60s seit dem letzten gelogten Wert verstrichen sind – im Logfile gespeichert und auf der Konsole ausgegeben.

Im zweiten Fall (keine 0x00000000 in $4 * count_warn$ Werten) wird das Programm beendet.

Der Parameter 'count_warn' ist deshalb nötig, weil die FIFOs in den verschiedenen Designs verschiedene Größen haben und so kontrolliert wird, ob ein Überlauf des entsprechenden FIFOs in der Zwischenzeit möglich war (eine entsprechende Warnung wird im Zweifelsfalle auf der Konsole ausgegeben und im Logfile gespeichert). Als Wert sollte hierbei jedoch nicht die FIFO-Größe angegeben werden sondern höchstens FIFO-Größe minus Zwei.

Geht ein Paket verloren – was angenommen wird, wenn die Antwort nicht in entsprechender Zeit einläuft – so wird der SCSN-Ring[Gar02] gewechselt. Kann die Kommunikation nicht in 10 Schritten wieder hergestellt werden, so wird das Programm beendet.

Alle Ausgaben des Programms erfolgen mit Zeitstempel, und alle Fehler werden (soweit möglich) zumindest im Logfile gespeichert.

A.4 Work-Test Design(s)

Das Work-Test Design ist für drei verschiedene FPGAs implementiert worden. Sie unterscheiden sich jedoch aufgrund der Größe der FPGAs auch in einigen Kleinigkeiten:

work_test_APEX : Zum Testen des auf der CIA-RORC[Ata] montierten EP20K400EFC672-2X. Dies ist das vollständige Design.

work_test_ACEX : Zum Testen des auf der ACEX-Karte[Ang01] montierten EP1K100QC108-2. Es enthält aus Platzgründen keinen Multiplizierer und auch nicht die dazu gehörigen Ausgangsregister.

work_test_EPXA1 : Zum Testen des auf dem EPXA1-Development-Board[ALTa] montierten EPXA1F484C1. Zusätzlich zum Multiplizierer fehlen hier – abgesehen von einem hinter dem Multiplexer – alle Datenregister.

Wie bei den Memory-Test Designs (Siehe A.3) funktioniert die Kommunikation mittels SCSN[Gar02]. Alle drei Work-Test Designs laufen mit derselben Software.

A.4.1 Das Work-Test FPGA Design

Das Work-Test Design emuliert auf dem FPGA eine Art mathematischen Coprozessor. Zu diesem Zweck werden die vom SCSN[Gar02] übertragene Adresse und die dazugehörigen Daten im Falle eines Schreibzugriffes etwas uminterpretiert. Die 16 Adressbits fungieren hierbei als Enable-Signale für 16 Schieberegister, wobei die unteren 16 Datenbits an jeweils einem der Eingänge liegen: (Ausschnitt aus 'wtest.c')

```
for(i=0;i<16;i++)
{
    __u16 tmp=(1<<i);
    if(Adr & tmp)
        Shiftregs[i]=(Shiftregs[i]<<1) | ((Dat & tmp) ? 1 : 0);
};
```

Die Ausgänge dieser Schieberegister können direkt via SCSN[Gar02] über die Adressen 0-15 gelesen werden. Außerdem sind an sie noch Logikblöcke angeschlossen, deren Ergebnisse über die Adressen 16-31 erreichbar sind.

Die einzelnen Ergebnisse ergeben sich wie folgt (Auszug aus 'wtest.c' mit leicht geänderter Formatierung):


```
Results[ 0] = Shiftregs[ 0];
Results[ 1] = ~Shiftregs[ 1];
Results[ 2] = Shiftregs[ 1] | Shiftregs[ 2];
Results[ 3] = Shiftregs[ 2] | Shiftregs[ 3];
Results[ 4] = Shiftregs[ 3] & Shiftregs[ 4];
Results[ 5] = Shiftregs[ 4] & Shiftregs[ 5];
Results[ 6] = Shiftregs[ 5] ^ Shiftregs[ 6];
Results[ 7] = Shiftregs[ 6] ^ Shiftregs[ 7];
Results[ 8] = Shiftregs[ 9] - Shiftregs[ 8];
Results[ 9] = Shiftregs[ 9] + Shiftregs[ 8];
Results[10] = Shiftregs[11] - Shiftregs[10];
Results[11] = Shiftregs[11] + Shiftregs[10];
Results[12] = Shiftregs[12]==0 ?
                Shiftregs[13] :
                Shiftregs[13] % Shiftregs[12];
Results[13] = Shiftregs[12]==0 ?
                -1 :
                Shiftregs[13] / Shiftregs[12];

Results[14] = withmult ?
                Shiftregs[13] * Shiftregs[14] :
                0;
Results[15] = withmult ?
                Shiftregs[14] * Shiftregs[15] :
                -1;
```

Der Parameter 'withmult' ist hierbei für das APEX-Design 'TRUE' bzw. 1 und für das ACEX- bzw. EPXA1-Design 'FALSE' bzw. 0. Die Register in der Auslese sind logisch gesehen überflüssig, da die Zeit zwischen zwei SCSN[Gar02]-Befehlen weit größer als die Laufzeiten ist.

A.4.2 'wtest.c'

Auch die Work-Test Software baut auf meinem SCSN-Interface (Siehe A.2) auf.

Parameter siehe Quellcode oder Fehlerausgabe.

Das Programm ruft den JamPlayer[?] auf, um das als erstes Argument übergebene Design über Port 0x378 (erster Parallelport) zu programmieren. Schlägt der Versuch mit entsprechendem Rückgabewert mehrfach fehl, so wird das Programm mit Fehler beendet. Das Programm sendet je Durchlauf mehrere Paare von Zufallszahlen als Adresse und Daten via SCSN[Gar02] an das FPGA und simuliert dessen Berechnungen in Software.

Kommt der Schreibbefehl nicht innerhalb einer bestimmten Zeit durch das Netzwerk, so wird das Programm mit Fehler beendet, da ein Schreibfehler in diesem Design nur schwer zu tolerieren ist.

Hiernach werden die Adressen 0-31 vom FPGA ausgelesen und mit den im PC berechneten Werten verglichen. Hierbei werden, wenn Fehler auftreten diese in der Logdatei sowie auf der Konsole gespeichert. In der Logdatei werden außerdem noch alle geschriebenen Daten und Adressen gespeichert. Das Programm beendet sich automatisch mit Fehler, wenn zuviele Fehler in einem Durchlauf auftreten.

Auf der Konsole wird außerdem alle 2^{14} Schreibbefehle eine Nachricht ausgegeben.

Alle Ausgaben des Programms erfolgen mit Zeitstempel, und alle Fehler werden (soweit möglich) zumindest im Logfile gespeichert.

A.5 ARM-Test Design

A.5.1 EPXA1 Design & Software

Bei dem ARM-Test Design handelt es sich direkt um das Beispieldesign von Altera[ALTC, ALTb]. Lediglich die Software wurde leicht verändert (Ausschnitte aus 'main.c', geänderte Formatierung):

```
void Scroll_LED(void)
{
    *LED<<=1;
    if(*LED==0) *LED=1;

    delay(10000000);

    return;
}
```

Hauptprogramm:

```
while (1)
{
    Scroll_LED();
    printf("%08X\r\n", *LED);
};
```

Bei diesem Design funktioniert die Kommunikation mit dem PC über eine RS232, eine normale serielle Schnittstelle. Allerdings ist keine Flusskontrolle mittels Xon-Xoff Protokoll implementiert.

A.5.2 'armlog.c'

Das Programm liest Daten von der als Parameter angegebenen seriellen Schnittstelle und gibt sie mit Zeitstempel sowohl auf der Konsole als auch in einem Logfile aus. Hierbei werden alle Daten, die zur Übernahme bereitstehen, als ein String behandelt. CR und LF (also '\r' und '\n') werden ignoriert.

B Danksagungen

Danken möchte ich vor allem den Leuten, die ihre eigene Zeit geopfert haben um mir bei meiner Arbeit helfend zur Seite zu stehen. Dazu zählen nicht nur meine Eltern und meine Schwester, sondern auch viele andere:

Beteiligten Professoren:

Prof. Dr. Volker Lindenstruth für das Thema der Arbeit, die Dienstreisen und die Unterstützung,

Prof. Dr. Norbert Herrmann für das Einverständnis als Zweitkorrektor zu fungieren, TI'ler:

Markus Gutfleisch für diverse Diskussionen,

Christian Reichling für Prozeßtechnische Unterstützung,

Dr. Venelin Angelov für Hilfe bei TRAP, Quartus und FPGA Designs,

Lord Hess für seine Kreditkarte zum Buchen der Dienstreisen,

Timm Steinbeck & Arne Wiebalk für Unterstützung bei Liunx & co.,

Deyan Atanasov für Hilfe bei Inbetriebnahme der CIA-Karte,

Robin Gareus & Jan deCuveland für L^AT_EX unterstützung,

Dr. Heiz Tilsner für Meßwerte,

Elektroniker:

Dirk Gottschalk, Holger Höbbel & Volker Kiworra für direkte & indirekte technische Unterstützung und Zuarbeit,

den restlichen für diverse Kabel und Löthilfe,

der Crew des OCL:

Jon Wikne & Eivind Atle Olsen mit technischer Unterstützung und Hilfe auch außerhalb der Strahlzeiten,

Externe:

Marc Stockmeier vom PI für diverse direkte & indirekte Hilfen sowie Diskussionen,

Ketil Røed (Uni Bergen) mit Hilfe in Oslo und am OCL,

...

Alle die glauben in diese Liste zu gehören und nicht auftauchen, bitte ich um Entschuldigung, daß ich vergessen habe sie aufzunehmen.

C Abbildungsverzeichnis

2.1	Typischer Aufbau eines MOS-FET	5
2.2	Aufbau eines CMOS Not Gatters	6
2.3	Aufbau einer SRAM Zelle	7
2.4	Aufbau einer DRAM Zelle	8
2.5	Aufbau eines FLASH-Transistors	9
2.6	Aufbau einer FLASH Speicherzelle	10
2.7	Kreuzung eines FPGA Verbindungsnetzwerks	11
3.1	Aufbau eines Thyristors	13
3.2	Zustandekommen parasitärer Thyristoren	14
4.1	Aufbau des Experiments bei der ersten und zweiten Strahlzeit	20
4.2	Aufbau des Experiments bei der dritten Strahlzeit	21
4.3	Aufbau in Natura	22
4.4	MT-SRAM bei $10 \frac{pA}{cm^2}$	24
4.5	MT-SRAM bei $50 \frac{pA}{cm^2}$	25
4.6	MT-SRAM bei $100 \frac{pA}{cm^2}$	26
4.7	Flußabhängigkeit der Fehlerrate in MT-SRAM	27
4.8	SDRAM bei $25 \frac{pA}{cm^2}$	30
4.9	SDRAM bei $50 \frac{pA}{cm^2}$	30
4.10	SDRAM bei $225 \frac{pA}{cm^2}$	31
4.11	Fehlerpositionen in SDRAM bei $50 \frac{pA}{cm^2}$	31
4.12	Fehlerpositionen in SDRAM modulo 8k bei $50 \frac{pA}{cm^2}$	32
4.13	Fehlerpositionen in SDRAM bei $225 \frac{pA}{cm^2}$	32
4.14	Fehlerpositionen in SDRAM modulo 8k bei $225 \frac{pA}{cm^2}$	33
4.15	Flußabhängigkeit der Fehlerrate in SDRAM (Einzelfehler)	34
4.16	Flußabhängigkeit der Fehlerrate in SDRAM („Block“-Fehler)	35
4.17	SDRAM bei $70 \frac{pA}{cm^2}$ (zweite Testreihe)	36
4.18	Flußabhängigkeit der Fehlerrate in SDRAM (zweite Testreihe)	37
4.19	Flußabhängigkeit der Fehlerrate in SDRAM (DCS-Board)	39
4.20	FLASH bei $1 \frac{nA}{cm^2}$	41

4.21 Fehlerposition im FLASH bei $1 \frac{nA}{cm^2}$	42
4.22 Fehlerwahrscheinlichkeit gegen Blockgröße im ACEX-FPGA (bei $10 \frac{pA}{cm^2}$)	45
4.23 Flußabhängigkeit der Lebensdauer des Work-Test Designs (APEX)	46
4.24 Flußabhängigkeit der Lebensdauer des Work-Test Designs (EPXA1 auf DCS-Board)	48
4.25 Flußabhängigkeit der ARM-Lebensdauer	50
4.26 Flußabhängigkeit der Fehlerrate in TRAP1 Eventbuffern	54
4.27 Flußabhängigkeit der MTTF in TRAP1 Eventbuffern	55
6.1 Schaltplan für die Simulation des SRAM Bitflips im TRAP Eventbuffer .	62
6.2 Simulation des SRAM Bitflips im TRAP Eventbuffer	64

D Akronyme

ALICE	„A Large Ion Colider Experiment“
ARM	„Advanced RISC Mashine“
CMOS	„Complementary Metal Oxide Semiconductor“
COTS	„Components Of The Shelf“
DCS	„Detector Control System“
DRAM	„Dynamic RAM“
EEPROM	„Electronically Eraseble and Programmable ROM“
EPI	„epitactical“
FIFO	„First In First Out“
FPGA	„Field Programable Gate Array“
IO	„Input/Output“
KIP	„Kirchhoff Institut für Physik“ Universität Heidelberg
LHC	„Large Hadron Colider“
MCM	„Multi Chip Module“
MOS-FET	„Metal Oxide Semiconductor Field Effect Transistor“
MTTF	„Mean Time To Failure“
MWPC	„Multi Wire Proportional Chamber“
NFS	„Network File System“
NIEL	„Non Ionising Energy Loss“
OCL	„Oslo Cyclotron Laboratory“ Universitetet i Oslo
PI	„Physikalisches Institut“Universität Heidelberg

D Akronyme

PSI.....	„PCI & Shared memory Interface“
RAM.....	„Read Access Memory“
SCSN	„Slow Control Serial Network“
SDRAM.....	„Synchrones DRAM“
SOI	„Silicon On Insulator“
SRAM.....	„Synchrones RAM“
TPC.....	„Time Projection Chamber“
TRAP	„Traclet Processor“
TRD.....	„Transition Radiation Detector“

E Literaturverzeichnis

- [ALTa] ALTERA. *EPXAI Development Board*.
<http://www.altera.com/literature/lit-exc.html>
- [AL Tb] *EPXAI Development Kit Design Files*.
<http://www.altera.com/literature/lit-exc.html>
- [AL Tc] ALTERA. *EPXAI Development Kit Getting Started User Guide*.
<http://www.altera.com/literature/lit-exc.html>
- [AL Td] ALTERA. *Excalibur Devices Hardware Reference Manual*.
<http://www.altera.com/literature/lit-exc.html>
- [AL Te] ALTERA. *PCI MegaCore Function User Guide*.
<http://www.altera.com/literature/megafunctions/lit-ipiup.html>
- [AMD] AMD. *AM29LV320D*. Datasheet.
- [Ang01] Venelin Angelov. *Acex Board*, 2001.
<http://www.kip.uni-heidelberg.de/ti/ACEXBoard/>
- [Ata] Deyan Atanasov. *CIA-Rorc Board Layout*.
- [BAE] *BAE-Systems*.
<http://www.iews.na.baesystems.com/space/asics/index.html>
- [CER01] CERN. *ALICE TRD Technical Design Report*. CERN/LHCC, 2001.
- [Eng] Christian Engwer. *PCI and Shared memory Interface*.
<http://ti.uni-hd.de/L3/documentation/software-and-documentation.html>
- [Fac] Federico Faccio. *COTS for the LHC radiation environment: the rules of the Game*.
http://lebwshop.home.cern.ch/lebwshop/LEB00_Book/plenary/faccio_plenary.pdf
- [Fri] Jeffrey E. F. Friedl. *Mastering Regular Expressions*. O'Reilly.

- [Gar02] Robin Gareus. *Slow Control - Serial Network and its implementation for the Transition Radiation Detector*. Diplomathesis, KIP, 2002.
<http://docserv.kip.uni-heidelberg.de/TI/>
- [H.V] H.Vogel. *Gerthsen Physik*. Springer. 19. revision.
- [lex] *dtv-Lexikon der Physik*.
- [Mic] Micron Technology. *Synchroneus DRAM 256Mb: x4, x8, x16 SDRAM*.
<http://www.micron.com/dramds/>
- [Pan] Ralph Panse. *memtest.c*.
- [Rho] Rhodes. *Imperfections and active centres in semiconductors*. International series of monographs on semiconductors.
- [Sto] Marc Stockmeier. *privat communications*.
- [Til] Heinz Tilsner. *memory-test software*.
- [Tip99] Till Toppel. *Investigations in radiation hard electronics*. Diplomathesis, Physikalisches Institut, 1999.
- [tJO] Larry Wall, Tom Christiansen & Jon Orwant. *Programming Perl*. O'Reilly.