

Modellierung und Steuerung der Temperaturverteilung Network-on-Chip-basierter Systeme

Dissertation
zur
Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
der Fakultät für Informatik und Elektrotechnik
der Universität Rostock



vorgelegt von
Wegner, Tim, geb. am 24.06.1984 in Leisnig
aus Rostock

Rostock, den 02. Januar 2014

Gutachter:

- Prof. Dr.-Ing. Dirk Timmermann, Universität Rostock
- Prof. Dr. rer. nat. habil. Adelinde M. Uhrmacher, Universität Rostock
- Prof. Dr.-Ing. Rainer Kokozinski, Universität Duisburg-Essen

Tag der Einreichung: 02. Januar 2014

Tag der Verteidigung: 07. Juli 2014

Danksagung

Die vorliegende Arbeit entstand während meiner wissenschaftlichen Tätigkeit am Institut für Angewandte Mikroelektronik und Datentechnik an der Universität Rostock.

An erster Stelle möchte ich meinem Doktorvater Prof. Dirk Timmermann danken, der diese Promotion überhaupt erst ermöglicht hat. Er sorgte neben der Anstellung am Institut für den Freiraum, der für die Fertigstellung dieser Dissertation essentiell war. Darüber hinaus sorgte er auch in schwierigen Phasen für die nötige Motivation und gab entscheidende Denkanstöße. In diesem Zusammenhang möchte ich auch meiner Betreuerin Prof. Adelinde Uhrmacher danken, welche meine Arbeit durch konstruktive Kritik bereicherte und mir stets mit Rat und Tat zur Seite stand.

Weiterhin möchte ich meiner Familie danken, die mich über die Jahre in allen Lebenslagen unterstützt und mir stets Kraft gegeben hat. Ohne sie und ihre Beständigkeit wäre diese Arbeit nie zustande gekommen.

Dank geht ebenfalls an meinen Kollegen und Bürogenossen Martin Gag, der mich tatkräftig unterstützt hat und durch die vielen wertvollen Diskussionen und Ratschläge für einen interessanten Arbeitsalltag gesorgt hat. Ein großes Dankeschön für fleißiges Korrekturlesen geht außerdem an die Kollegen Martin Gag, Peter Danielis und Philipp Gorski. Darüber hinaus danke ich dem gesamten Team des Instituts für die angenehme und harmonische Arbeitsatmosphäre und die schöne Zeit während meines Studiums.

Inhaltsverzeichnis

Abbildungsverzeichnis	VII
Tabellenverzeichnis	IX
Abkürzungsverzeichnis	XI
1 Einleitung	1
2 Ausgewählte Schwerpunkte hochintegrierter Systeme	5
2.1 Technologische Aspekte	5
2.1.1 Entwicklung der Fertigungstechnologie	5
2.1.2 Kritische Faktoren	6
2.2 Zuverlässigkeit	10
2.2.1 Grundlagen und Definitionen	10
2.2.2 Ausfallursachen	12
2.2.3 Einfluss der Temperatur	15
2.2.4 Existierende Techniken zur Steigerung der Zuverlässigkeit	20
2.3 Kommunikationsstrukturen nanoelektronischer Systeme	25
2.3.1 Überblick und analytischer Vergleich	26
2.3.2 Grundlagen NoC-basierter Systeme	31
2.3.3 NoC-Topologien	38
2.3.4 Zuverlässigkeit NoC-basierter Systeme	42
2.4 Ermittlung der Temperatur hochintegrierter Schaltungen	47
2.4.1 Messung mithilfe von Ringoszillatoren	48
2.5 Folgerung und Problemformulierung	53
3 Temperaturmodell	57
3.1 Realisierung	57
3.1.1 Dualismus elektrischer und thermischer Energie	57
3.1.2 Stand der Technik	59
3.1.3 Implementierung und Simulationsablauf	60
3.2 Bewertung des Modells	70
3.2.1 Zeitverhalten	72
3.2.2 Korrektheit des Modells	78
3.2.3 Qualitative Bewertung und Zusammenfassung	84
4 Temperaturorientiertes Management NoC-basierter Systeme	89
4.1 Stand der Technik	89
4.2 Anwendungsszenario I: Proaktives Temperaturmanagement	94
4.2.1 Konzept	95

4.2.2	Vergleich mit reaktivem Management und Bewertung	108
4.2.3	Zusammenfassung	121
4.3	Anwendungsszenario II: Proaktives Task-Mapping	124
4.3.1	Konzept	124
4.3.2	Vergleich mit gängigen Konzepten und Bewertung	128
4.3.3	Zusammenfassung	134
5	Zusammenfassung und Ausblick	137
5.1	Zusammenfassung	137
5.2	Ausblick	140
Literaturverzeichnis		143

Abbildungsverzeichnis

1.1	Struktur der Arbeit	4
2.1	Die in einer digitalen Schaltung auftretenden Lade- und Kurzschlussströme am Beispiel eines Inverters	8
2.2	Einfluss der Fertigungstechnologie auf die Transistor- und Verlustleistungsdichte	9
2.3	Lebenszyklus hochintegrierter Schaltungen	11
2.4	Klassifikation der Ursachen für das Fehlverhalten hochintegrierter Schaltungen	13
2.5	Einfluss eines Temperaturanstiegs auf die Time To Failure	19
2.6	Kategorisierung der Strategien zur Erhöhung der Zuverlässigkeit	22
2.7	Design-Productivity Gap	25
2.8	Wichtige Kommunikationsarchitekturen: Punkt-zu-Punkt-Verbindung, Bus und NoC	28
2.9	Aufbau einer Nachricht	32
2.10	Aufbau eines NoC-Routers	33
2.11	NoC-Netzwerkschnittstelle	34
2.12	Klassifikation der Routing-Algorithmen	36
2.13	Gitternetz: 2D-Gitter und 3D-Gitter	39
2.14	Torus: 3-facher 2-Würfel und Ring	40
2.15	Grad der Vernetzung: durchschnittliche Anzahl der Übertragungskanäle je Knoten und Gesamtanzahl der Übertragungskanäle	41
2.16	Beispiel für XY-Routing und West-First-Routing	44
2.17	Auswirkung von Router-Ausfällen auf die Zuverlässigkeit und Leistungsfähigkeit	45
2.18	Auswirkung des Ausfalls von Übertragungskanälen auf die Zuverlässigkeit und Leistungsfähigkeit	47
2.19	Schematischer Aufbau eines Ringoszillators	49
2.20	Platzierung der Temperatursensoren und Test-Heater auf dem FPGA . . .	50
2.21	Temperaturausbreitung auf dem FPGA	52
2.22	Dauer der Temperaturausbreitung für verschiedene Zonen des FPGA . . .	53
2.23	Allgemeiner Aufbau und Einordnung eines mehrschichtigen, zuverlässigkeitssorientierten Systemmanagements	55
3.1	Äquivalenz zwischen elektrischem und thermischem Widerstand	59
3.2	Abbildung eines 2×2 NoC auf ein äquivalentes RC-Netz	62
3.3	Genauigkeitsstufen der Abbildung eines 2×2 NoC auf ein äquivalentes RC-Netz	65
3.4	Vollständig modellierter Chip	66
3.5	Ablauf der Simulation des Temperaturverlaufs	68

3.6	Beispiel für ein mithilfe von ELN-Primitiven modellierbares elektrisches Netzwerk	69
3.7	Modellierung einer Stromquelle mit SPICE und SystemC AMS	72
3.8	Zeitaufwand der Temperatursimulation in Abhängigkeit von der NoC-Größe	76
3.9	Zeitaufwand der Temperatursimulation in Abhängigkeit von T_S und T_A	77
3.10	Ermittelte durchschnittliche Temperatur im System	79
3.11	Modellierung des Temperaturprofils eines 2×2 NoC	80
3.12	Ermittelte Temperatur eines einzelnen Routers unter Verwendung verschiedener Auflösungen	81
3.13	Ermittelte Temperatur eines einzelnen Routers unter Verwendung verschiedener Simulationsperioden	83
4.1	Schema der Umgebung zur Simulation des proaktiven Temperaturmanagements NoC-basierter Mehrkernprozessoren	97
4.2	Interne Funktionsweise des proaktiven Temperaturmanagements	98
4.3	Einteilung eines 2×2 NoC in Zuständigkeitsbereiche einzelner Sonden . .	100
4.4	Alternativen eine Sonde an das NoC anzuschließen	102
4.5	Schema der Umgebung zur Simulation des reaktiven Temperaturmanagements NoC-basierter Mehrkernprozessoren	104
4.6	Interne Funktionsweise des reaktiven Temperaturmanagements	105
4.7	Exemplarische Kommunikation zwischen Sonde und TMU	106
4.8	Ergebnisse für das proaktive sowie das reaktive Temperaturmanagement bezüglich der Durchschnittstemperatur, der Maximaltemperatur und der maximalen Temperaturunterschiede in einem 4×4 NoC	111
4.9	Verlauf der Maximaltemperatur in einem 4×4 NoC über die Dauer einer Sekunde	112
4.10	Durchschnittswerte für die Dauer einer Paketübertragung und die Verzögerung der Weiterleitung eines Flits durch einen Router in einem 4×4 NoC für das proaktive und das reaktive Temperaturmanagement	114
4.11	Durchschnittliche Dauer der durch DFS reduzierten Taktfrequenz der IPCs und Router in einem 4×4 NoC für das proaktive und das reaktive Temperaturmanagement	116
4.12	Durchschnittliche Netto-Datenrate in einem 4×4 NoC für das proaktive und das reaktive Temperaturmanagement	118
4.13	Anwendungsmodellierung durch einen Task-Graphen	125
4.14	Umgebung zur Simulation des proaktiven Task-Mappings NoC-basierter Mehrkernprozessoren	127
4.15	Ergebnisse bezüglich der Durchschnittstemperatur, der Maximaltemperatur und der maximalen Temperaturunterschiede in einem 4×4 NoC	131
4.16	Anzahl fertiggestellter Tasks und durchschnittliche Leerlaufdauer der Systemkomponenten in einem 4×4 NoC	133

Tabellenverzeichnis

2.1	Qualitative Bewertung und Vergleich Punkt-zu-Punkt-basierter, busbasierter und NoC-basierter Kommunikationsarchitekturen	30
2.2	Parameter für die Simulation der Auswirkung von Ausfällen auf gitternetz- und torusbasierte NoCs	43
2.3	Genauigkeit der ringoszillatortbasierten Temperatursensoren	51
3.1	Dualismus zwischen dem Fluss elektrischer und thermischer Energie . . .	58
3.2	Energieaufnahme je Bitübergang sowie statische Leistungsaufnahme für einzelne Router-Komponenten, den IPC und den Übertragungskanal . . .	64
3.3	Simulationsparameter für die Verifikation des Temperaturmodells NoC-basierter Systeme	71
3.4	Zeitaufwand für die Temperatursimulation eines 2×2 NoC	75
3.5	Abweichung der durchschnittlichen Temperatur des Systems vom SPICE-Referenzmodell für verschiedene Simulationsperioden	82
3.6	Qualitative Bewertung des implementierten Temperaturmodells	86
4.1	Qualitative Bewertung der Anbindungsalternativen einer Sonde	103
4.2	Parameterkonfigurationen des proaktiven und reaktiven Temperaturmanagements	109
4.3	Ergebnisse für die Anzahl der Überwachungsnachrichten sowie der Steuerungsinstruktionen für das proaktive und das reaktive Temperaturmanagement in einem 4×4 NoC	121
4.4	Durchschnittliche Temperatur in NoCs verschiedener Größe unter Verwendung der Standardkonfiguration K1.1	122
4.5	Qualitative Bewertung des proaktiven Temperaturmanagements	123
4.6	Qualitative Bewertung des proaktiven Task-Mappings	135
5.1	Abschätzung der Auswirkung einer Kombination der proaktiven Konzepte für Temperaturmanagement und Task-Mapping	142

Abkürzungsverzeichnis

AMS	Analog Mixed Signal
AP	Application Processor
ARQ	Automatic Repeat Request
BE	Best Effort
BFT	Butterfly Fat Tree
BIST	Built In Self Test
CAD	Computer-Aided Design
CLB	Configurable Logic Block
CMOS	Complementary Metal Oxide Semiconductor
CRC	Cyclic Redundancy Check
DFS	Dynamic Frequency Scaling
DGS	Differentialgleichungssystem
DPG	Design-Productivity-Gap
DVS	Dynamic Voltage Scaling
E/A	Ein-/Ausgabe
EDP	Energy-Delay-Product
ELN	Electrical Linear Network
EM	Elektromigration
FET	Field Effect Transistor
FIFO	First In First Out
Flit	Flow Control Unit
FPGA	Field Programmable Gate Array
GPP	General Purpose Processor
GPU	Graphics Processing Unit
GS	Guaranteed Service
HCI	Hot Carrier Injection
IPC	Intellectual Property Core
ITRS	International Technology Roadmap for Semiconductors
LUT	Lookup Table
MOSFET	Metal Oxide Semiconductor Field Effect Transistor

MPSoC	Multi-Processor-System-on-Chip
MTTF	Mean Time To Failure
MuGFET	Multiple Gate Field Effect Transistor
NBTI	Negative Bias Temperature Instability
NFR	Negative-First-Routing
NLR	North-Last-Routing
NoC	Network-on-Chip
PDP	Power-Delay-Product
Phit	Physical Transfer Unit
PWL	Piecewise Linear Source
QoS	Quality-of-Service
RA	Routing-Algorithmus
RDNI	Resource Dependent Network Interface
RINI	Resource Independent Network Interface
RO	Ringoszillator
RTL	Register Transfer Level
SAFS	Store-And-Forward Switching
SNR	Signal-to-Noise Ratio
SoC	System-on-Chip
SoI	Silicon-on-Insulator
SPICE	Simulation Program with Integrated Circuit Emphasis
TDDB	Time Dependent Dielectric Breakdown
TDP	Thermal Design Power
TG	Task-Graph
TGFF	Task Graphs For Free
TLM	Transaction Level Modeling
TMR	Triple Modular Redundancy
TMU	Thermal Management Unit
TTF	Time To Failure
VCTS	Virtual-Cut-Through Switching
WFR	West-First-Routing
WHS	Wormhole Switching
XYR	XY-Routing

1 Einleitung

Die fortschreitende Miniaturisierung hochintegrierter Schaltkreise und die Einführung von Fertigungstechnologien mit Strukturgrößen weit im Submikrometerbereich haben neben einer Erhöhung der Anzahl integrierbarer Transistoren bei gleichbleibender Fläche, und damit einer Erhöhung der Leistungsfähigkeit moderner Mikrochips, auch negative Effekte. So stößt man zunehmend an die Grenzen des technisch Machbaren und wird mit wachsenden Problemen wie stark zunehmenden Leckströmen und steigenden Leistungsdichten konfrontiert.

Zielstellung Die Entwicklung moderner Fertigungstechnologien in der Halbleitertechnik bis in den Bereich weniger Nanometer korrespondiert mit dem hinlänglich bekannten Moore'schen Gesetz, welches von Gordon E. Moore 1965 postuliert wurde [Moo65]. Auch für die nähere Zukunft prognostiziert die International Technology Roadmap for Semiconductors (ITRS) in ihrem Bericht für 2012 in den nächsten 10 bis 12 Jahren eine weitere Verkleinerung der Strukturgrößen bis unterhalb der Grenze von 10 nm [ITR12]. Dies soll, trotz der sich immer schwieriger gestaltenden physikalischen Bedingungen, durch den Einsatz neuer Materialien und Strukturen erreicht werden. In den kommenden Jahren werden vorwiegend Modifikationen klassischer CMOS-Strukturen genutzt, welche sich beispielsweise der Effekte von Materialien mit erhöhter Durchlässigkeit für elektrische Felder (sogenannte „high-k“-Materialien) bedienen [Rob06]. Längerfristig werden die in herkömmlicher Planartechnik gefertigten CMOS-Technologien allerdings durch stark modifizierte oder vollständig neue Technologien ergänzt und letztendlich ersetzt. Beispielhaft seien hier spezielle Silicon-on-Insulator-Techniken (SoI) [SIC10] sowie Multiple Gate Field Effect Transistors (MuGFETs) [Ris05] (z. B. Tri-Gate-FETs oder FinFETs) angeführt. Trotz des Einsatzes solch neuartiger Verfahrens- und Herstellungstechniken, und der damit verbundenen Reduzierung von Verlustleistung und Leckströmen, ist eine zunehmende Anfälligkeit hochintegrierter Schaltungen gegenüber Umwelteinflüssen sowie Alterungs- und Verschleißerscheinungen zu erwarten.

Des Weiteren führt die nur begrenzt steigerbare Leistungsfähigkeit einzelner Prozessoren dazu (siehe auch Pollack's Law [Bor07]), dass der benötigte Leistungszuwachs durch ein wachsendes Maß an Parallelität kompensiert werden muss. Das Resultat sind vorwiegend heterogene Multi-Processor-Systems-on-Chip (MPSoCs) sowie Manycore-Prozessoren [Til12a, Ada12a, Int12], welche derzeit bis zu 256 [Kal12] in der Regel ho-

mogene Rechenkerne auf einem einzelnen Chip vereinen und damit in der Lage sind, parallel mehrere Aufgaben abzuarbeiten. Die Kommunikation zwischen den Rechenkernen wird aufgrund verschiedener Vorzüge derartiger Konzepte durch netzwerkorientierte Architekturen wie Networks-on-Chip (NoCs) realisiert [Til12b, Ada12b, Art12, Son12]. Durch die wachsende Parallelität und steigende Komplexität wird dabei der Verwaltungs- und Koordinationsaufwand für ein System-on-Chip (SoC) stark erhöht. Dieser Effekt wird durch sogenannte „Multi-Everything“-Ansätze zusätzlich verstärkt. Diese ermöglichen zwar erst eine leistungs- oder energieoptimierte Steuerung und setzen hierfür beispielsweise multiple Versorgungsspannungen oder Betriebsfrequenzen ein. Jedoch erhöht sich durch diese Diversifizierung auch der erforderliche Aufwand.

Zusätzlich haben die mit neuen Fertigungstechnologien schrumpfenden Strukturgrößen immer filigranere und somit generell empfindlichere Strukturen zur Folge. Dies führt zudem zu einer erhöhten Packungsdichte der integrierten Transistoren und damit zu einer steigenden Dichte der thermischen Verlustleistung, da auf gleichem Raum eine zunehmende Anzahl von Komponenten Leistung verbraucht. Zwar stagniert der Wert der Verlustleistungsdichte seit Einführung der 130 nm-Fertigung, hält aber ein vergleichsweise hohes Niveau. Dies ist insofern als kritisch zu bewerten, dass somit temperaturbezogene Probleme stärker denn je an Bedeutung gewinnen. Neben thermischem Stress durch Temperaturspitzen und -gefälle existieren einige temperaturabhängige physikalische Effekte, welche durch hohe Temperaturen begünstigt werden. Zu diesen Effekten zählen beispielsweise Time Dependent Dielectric Breakdown (TDDB), Negative Bias Temperature Instability (NBTI) oder Elektromigration (EM), welche maßgeblich zu einer beschleunigten Alterung hochintegrierter Schaltkreise beitragen.

Aus den vorangegangenen Überlegungen ergeben sich Motivation und Zielstellung der vorliegenden Arbeit. Deren Hauptaugenmerk liegt auf der Abschwächung beziehungsweise der Verzögerung von Alterungs- und Verschleißerscheinungen. Aufgrund des zunehmenden Einflusses thermischer Effekte wird insbesondere untersucht, inwiefern das frühzeitige Erkennen von Temperaturveränderungen und Temperaturtrends zu einer Vorbeugung von Temperaturspitzen und zu einer langfristigen Reduzierung thermischen Stresses beitragen kann. Diese Arbeit konzentriert sich aufgrund der Komplexität des Feldes der On-Chip-Kommunikationsarchitekturen dabei ausschließlich auf NoC-basierte Architekturen, zumal sich diese durch eine hohe Flexibilität und eine hervorragende Skalierbarkeit als besonders vielversprechend hervorheben. Die entstehenden proaktiven Konzepte sollen neben einer Verbesserung des Temperaturprofils durch vorausschauendes Anpassen von Lastverteilung und abrufbarer Leistung möglichst auch eine Verringerung der unumgänglichen Reduzierung der Leistungsfähigkeit bewirken.

Struktur der Arbeit Die vorliegende Arbeit unterteilt sich in fünf Kapitel. Diese Struktur ist in Abbildung 1.1 dargestellt. Die dunkel hervorgehobenen Abschnitte enthalten die wesentlichen eigenen wissenschaftlichen Beiträge.

Kapitel 1 gibt eine kurze Einführung in die Thematik der Arbeit. Dabei werden aktuelle technologische Entwicklungen und Problemstellungen umrissen, welche maßgeblich zur Motivation und Zielstellung dieser Arbeit beitragen. Zusätzlich wird ein Überblick über die Struktur der Arbeit sowie darin enthaltene eigene Beiträge gegeben.

In Kapitel 2 wird eine Reihe thematischer Schwerpunkte behandelt, welche für diese Arbeit relevant sind. Dazu werden eingehende Betrachtungen zu den Bereichen technologischer Aspekte und der Architektur moderner nanoelektronischer Systeme sowie Probleme und Maßnahmen deren Zuverlässigkeit betreffend angestellt. Weiterhin werden Ansätze zur Ermittlung der Temperatur hochintegrierter Schaltungen vorgestellt, wobei eine für diese Arbeit bedeutsame Methode näher untersucht wird.

In Kapitel 3 wird ein Modell vorgestellt, mit dessen Hilfe eine akkurate und schnelle Prognose kurzfristiger Temperaturänderungen NoC-basierter Systeme zur Laufzeit getroffen werden kann. Dieses Modell wird im Hinblick auf Genauigkeit und Leistungsfähigkeit untersucht und bezüglich seiner Tauglichkeit für ein proaktives Systemmanagement bewertet.

Kapitel 4 befasst sich mit der Steuerung und Überwachung NoC-basierter Systeme. Zunächst wird ein Überblick über den Stand der Technik bezüglich gängiger Methoden für das Management nanoelektronischer Systeme gegeben. Im Anschluss wird das in Kapitel 3 vorgestellte Temperaturmodell für zwei verschiedene Ansätze einer proaktiven temperaturorientierten Systemsteuerung genutzt. Ein Vergleich mit etablierten Mechanismen ermöglicht Aussagen bezüglich des Mehrwertes und der Einsatzmöglichkeiten der vorgestellten Ansätze. Die gewonnenen Erkenntnisse und daraus resultierende Schlussfolgerungen werden abschließend zusammengefasst.

Kapitel 5 fasst die in dieser Arbeit gewonnenen Erkenntnisse nochmals zusammen, diskutiert bestehende Probleme und gibt einen Ausblick auf weitere Forschungsschwerpunkte.



Teilweise eigene
Beiträge



Eigene Beiträge

Abbildung 1.1: Struktur der Arbeit: Abschnitte, die eigene wissenschaftliche Beiträge enthalten, sind dunkel gekennzeichnet

2 Ausgewählte Schwerpunkte hochintegrierter Systeme

Diese Arbeit setzt sich mit der Zuverlässigkeit nanoelektronischer Systeme auseinander und konzentriert sich dabei auf ein temperaturoptimiertes Management NoC-basierter Mehrkernprozessoren. Um die Nachvollziehbarkeit und Verständlichkeit der Arbeit zu gewährleisten, wird im Folgenden auf die notwendigen theoretischen Grundlagen eingegangen. Zu diesen zählen neben der Betrachtung allgemeiner technologischer Aspekte sowie der Hervorhebung damit verbundener kritischer Faktoren vor allem eine Einführung in die Architektur nanoelektronischer Systeme, ein Überblick über den Einfluss physikalischer Effekte auf deren Zuverlässigkeit sowie ein Abriss bezüglich der Möglichkeiten zur Ermittlung der Temperatur hochintegrierter Schaltungen.

2.1 Technologische Aspekte

Aktuelle Mikroprozessoren werden mit Halbleitern gefertigt, deren kleinste fotolithografisch herstellbare Strukturgröße 22 nm beträgt. Auch zukünftig ist durch neue Fertigungstechnologien mit weiteren Reduzierungen der geometrischen Dimensionen sowie der Spannungsspegel hochintegrierter Schaltungen zu rechnen [ITR13b]. Dies führt neben einem verringerten Platzbedarf und somit der Möglichkeit, mehr Komponenten auf einem einzelnen Chip zu integrieren, sowie einer Reduzierung der Leistungsaufnahme auch zu unerwünschten Effekten.

2.1.1 Entwicklung der Fertigungstechnologie

Im Bereich der mikro- und nanoelektronischen Systeme ist ein stetig steigender Bedarf an Flexibilität und Leistungsfähigkeit zu verzeichnen. Dies manifestiert sich in einer steigenden Anzahl der auf einem einzelnen Chip integrierten Systemkomponenten sowie einer zunehmenden Heterogenität dieser Komponenten [KTJR05, CLP⁺09]. Dieser Trend wird besonders im Sektor der mobilen Geräte deutlich, zu denen beispielsweise Smartphones, Tablets und PDAs zählen. Demgegenüber stehen steigende Anforderungen bezüglich der Zuverlässigkeit und der Energieeffizienz sowie der Time-to-Market, der Produktqualität und der Kostenminimierung [ITR12]. Nicht zuletzt durch diese gegensätz-

liche Entwicklung ist ein Paradigmenwechsel im Bereich der On-Chip-Kommunikation zu beobachten. Dieser führt dazu, dass herkömmliche, busbasierte Kommunikationsarchitekturen durch leistungsfähigere, modulare und besser skalierbare Konzepte ersetzt werden.

Die steigende Systemkomplexität erfordert einen erhöhten Planungs- und Koordinationsaufwand, um die zusätzlich verfügbare Leistung effizient nutzen und einen reibungslosen Ablauf gewährleisten zu können. Die Ursache hierfür liegt in einer höheren Anzahl parallel arbeitender und untereinander kommunizierender Komponenten. Diese weisen neben unterschiedlichen Aufgabenprofilen unter Umständen auch verschiedene Priorisierungen, zeitliche Vorgaben und weitere Rahmenbedingungen auf. Solche Komponenten (auch Rechenkern oder Ressource) lassen sich in die fünf Kategorien der Mehrzweckprozessoren (General Purpose Processors (GPP)), der anwendungsspezifischen Prozessoren (Application Processors (AP)), der Beschleunigungslogik (Acceleration Logic), der Speicherkomponenten sowie der Ein- und Ausgabe (E/A) einteilen.

Ein weiterer kritischer Faktor ist der erhöhte Regulierungsaufwand für die On-Chip-Kommunikation zwischen der wachsenden Anzahl von Komponenten. Die Heterogenität der On-Chip-Komponenten hat zudem zur Folge, dass auch der Datenverkehr selbst heterogen ist (z. B. Echtzeit- oder Streaming-Daten, CPU-Instruktionen, Steuerungsinformationen) und dementsprechend komplexere Abarbeitungsmechanismen erforderlich sind. Zusätzlich sorgen die aufgeführten Effekte dafür, dass sich eine etwaige Systemüberwachung und -steuerung aufwändiger und komplexer gestaltet. Zur Überwachung gehört beispielsweise das Erstellen von Verkehrsstatistiken oder die Analyse einzelner Pakete. Die Steuerung hingegen beinhaltet Wartung, Selbstheilung, Fehlerdetektion und -korrektur sowie die Anpassung des Systems an vorgegebene Last-, Temperatur- oder Energieprofile.

Neben diesen eher auf System- beziehungsweise Entwurfsebene gelegenen Problemfeldern existiert auch eine Reihe viel grundlegenderer teils physikalischer Faktoren. Diese gehen mit der Weiterentwicklung der Fertigungstechnologien einher und gewinnen gerade im Bereich der MPSoCs und Mehrkernprozessoren zunehmend an Bedeutung. Hierzu zählen unter anderem Aspekte wie Energieminimierung und Leistungsdichte sowie temperaturbezogene Probleme wie Temperaturspitzen und thermischer Stress.

2.1.2 Kritische Faktoren

Aufgrund der immer filigraneren Halbleiterstrukturen steigt sowohl die absolute Anzahl leistungsverbrauchender Komponenten als auch deren Dichte je Flächeneinheit (siehe Abbildung 2.2), da einzelne Elemente kompakter ausgelegt werden können. Dies hat einerseits zur Folge, dass die Anfälligkeit der nanoelektronischen Bauteile gegenüber Umwelteinflüssen und Alterungserscheinungen steigt, wodurch sich die effektive Le-

bensdauer einer Schaltung potentiell verringert. Zudem erhöht sich durch die steigende Anzahl von Transistoren die Wahrscheinlichkeit eines Defekts und eines entsprechenden Ausfalls. Andererseits ermöglichen kleinere Fertigungsstrukturen eine Reduzierung der Gesamtleistungsaufnahme hochintegrierter Schaltungen. Diese Reduzierung ist angesichts einer steigenden Anzahl zu versorgender Elemente erforderlich, um den Energiebedarf digitaler Schaltungen zu begrenzen. Dabei wird ausgenutzt, dass sich im Zuge der Strukturverkleinerungen ebenfalls die Gate-Oxid-Dicke der Transistoren verringern lässt. Damit wird ein Absenken der Versorgungsspannung V_{DD} ermöglicht [ITR13b], was sich positiv auf alle Komponenten der Leistungsaufnahme auswirkt.

Diese setzt sich aus der dynamischen Leistungsaufnahme P_{Dyn} , der statischen Leistungsaufnahme P_{Leck} und der durch Kurzschlüsse bedingten Leistungsaufnahme P_K zusammen (siehe Gleichung 2.1). P_{Dyn} ist der Anteil der Leistung, der aufgenommen wird, solange eine Schaltung aktiv arbeitet, also während eine Lastkapazität C_L aufgeladen wird. Am Beispiel eines Inverters (siehe Abbildung 2.1) ist dies der Fall während der Ladestrom I_L fließt. P_K ist die Leistung, welche durch einen temporären Kurzschluss zwischen V_{DD} und Masse einer Schaltung aufgenommen wird. Für das Beispiel des Inverters bedeutet dies, dass für die Dauer von t_K beide Transistoren leiten und der Kurzschlussstrom I_K fließt (siehe Abbildung 2.1). Die Wahrscheinlichkeit, dass ein Kurzschluss auftritt ist dabei doppelt so hoch wie für P_{Dyn} , da sich bei jedem Pegelwechsel des Eingangssignals ein leitender Pfad bilden kann. P_{Leck} wird im Gegensatz zu den anderen beiden Leistungskomponenten fortwährend aufgenommen, also auch während die Schaltung inaktiv ist. Der Leckstrom I_{Leck} setzt sich aus einer Reihe einzelner Ströme mit unterschiedlichen, zum Teil technologisch bedingten Ursachen zusammen. Die drei wichtigsten Ströme sind der Subthreshold-Leckstrom I_{Sub} , der Gate-Oxid-Leckstrom I_{Gate} sowie der Leckstrom am pn-Übergang eines Transistors I_{pn} . I_{Sub} fließt zwischen Drain und Source eines Transistors während dieser theoretisch sperren sollte. I_{Gate} ist auf Ladungsträger zurückzuführen, welche aufgrund von Tunneleffekten in das Gate-Oxid eines Transistors gelangen. I_{pn} wird durch Diffusionsströme am pn-Übergang eines Transistors verursacht.

$$P_{Gesamt} = \underbrace{\alpha \cdot f \cdot C_L \cdot V_{DD}^2}_{P_{Dyn}} + \underbrace{2 \cdot \alpha \cdot f \cdot t_K \cdot I_K \cdot V_{DD}}_{P_K} + \underbrace{\sum I_{Leck} \cdot V_{DD}}_{P_{Leck}} \quad (2.1)$$

Das α bezeichnet die Schaltwahrscheinlichkeit, C_L die umzuladende Lastkapazität (siehe auch Abbildung 2.1), f die Taktfrequenz der betrachteten Schaltung und t_K die Dauer des Kurzschlussstroms. I_K und I_{Leck} kennzeichnen die auftretenden Kurzschlussbeziehungsweise Leckströme.

Die Reduzierung der Leistungsaufnahme durch ein Absenken von V_{DD} ist allerdings durch eine minimale Schichtdicke von wenigen Atomlagen (ca. 1 bis 2 nm) des Gate-

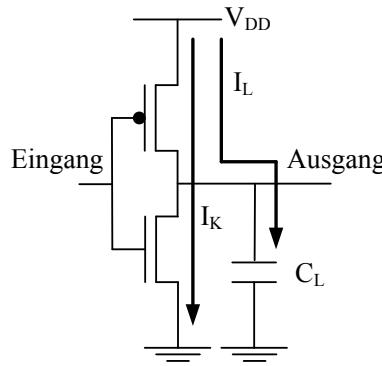


Abbildung 2.1: Die in einer digitalen Schaltung auftretenden Lade- und Kurzschlussströme I_L und I_K am Beispiel eines Inverters

Oxids beschränkt. Andernfalls kommt es zur Verstärkung der auftretenden Leckströme [LBS04]. Die auf diese Weise erzielte Einsparung hat zudem eine reduzierte Signalintegrität zur Folge, da ein Absenken des V_{DD} -Pegels einen geringeren Signal-Rausch-Abstand (SNR) verursacht. Zudem darf sich V_{DD} nicht zu sehr dem Wert der Schwellspannung V_{Th} nähern, welche den für das Umschalten eines Transistors nötigen Spannungspegel angibt. Andernfalls verringert sich die Geschwindigkeit der Umladung der Lastkapazitäten, da ein geringerer Lade- beziehungsweise Entladestrom fließt.

Daher muss im gleichen Maße auch V_{Th} herabgesetzt werden, um das Verhältnis zwischen V_{DD} und V_{Th} zu wahren und ein korrektes Schaltverhalten des Transistors zu gewährleisten. Dies hat allerdings steigende Leckströme und somit eine erhöhte Leistungsaufnahme zur Folge, sodass letztendlich ein Kompromiss zwischen der Beeinflussung des Schaltverhaltens und der Kompensation der auftretenden Leckströme angestrebt werden muss [ITR13b].

Beschleunigte Umladevorgänge führen neben einer erhöhten Leistungsfähigkeit aber auch zu einer steigenden Belastung der Transistoren, da die Anzahl der Ladevorgänge je Zeiteinheit bei geringerer Belastbarkeit der Bauteile entweder konstant bleibt oder sogar zunimmt. Des Weiteren erhöht sich durch die zusätzlichen Schalt- und Umladevorgänge die Menge der abgegebenen Wärme, was wiederum einen negativen Einfluss auf V_{Th} hat [GWWT12] und zur Erhöhung der thermischen Verlustleistung beiträgt.

Die Auswirkung technologiebedingter Strukturverkleinerungen auf die Leistungsdichte ist in Abbildung 2.2 dargestellt. Wie erkennbar ist, erhöht sich die Dichte der thermischen Verlustleistung mit jeder Strukturverkleinerung bis 130 nm und stagniert von dort an auf konstant hohem Niveau. Die Dichte bezieht sich auf die Thermal Design Power (TDP), welche die maximal von einem Prozessor abgegebene Wärmeleistung angibt. Das hohe Niveau der TDP ist dabei eher praktischen Erwägungen als physikalischen Beschränkungen geschuldet. Durch effektivere, aber ungleich aufwändiger und

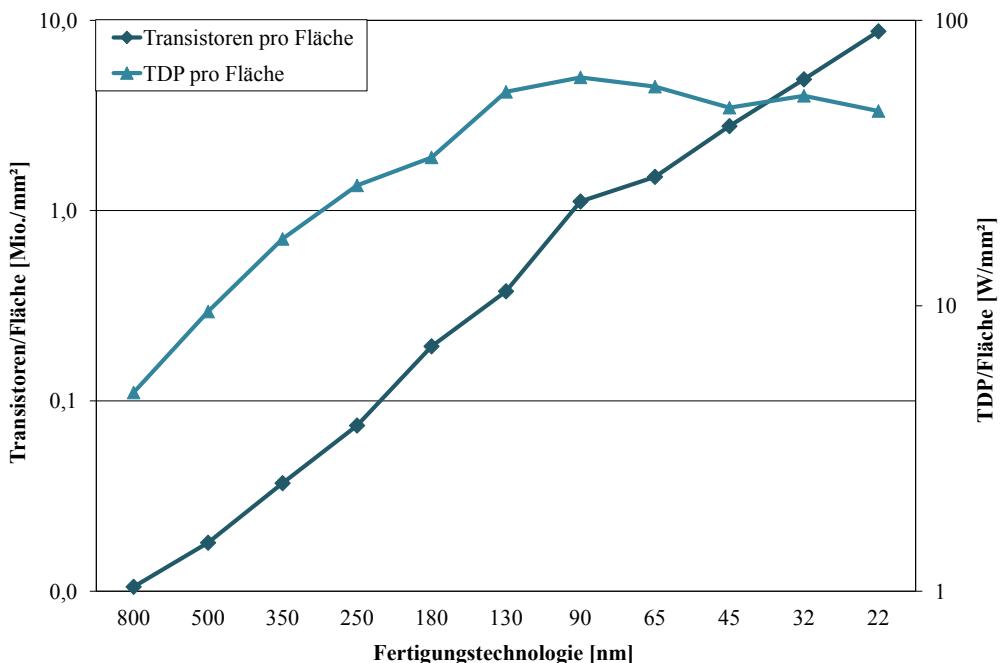


Abbildung 2.2: Einfluss der Fertigungstechnologie auf die Transistor- und Verlustleistungsdichte (Werte beruhen auf den Daten für Intel-Prozessoren der jeweiligen Technologiegeneration)

kostspieligere, Kühlösungen kann die Menge der abführbaren Wärme jederzeit erhöht werden. Ungeachtet dessen verdeutlicht der Verlauf dieser Kurve, dass sich die Intensität der Wärmewirkung auf einzelne Bauteile und Transistoren im Zuge technologischer Weiterentwicklungen stark erhöht hat. Hochintegrierte Schaltungen sind somit stetig steigenden Temperaturen und zunehmendem thermischen Stress ausgesetzt. Eine detaillierte Betrachtung bezüglich des Einflusses der Temperatur auf verschiedene Parameter mikro- und nanoelektronischer Schaltungen und daraus resultierender Effekte findet sich in Abschnitt 2.2.3.

Die vorangegangenen Betrachtungen zeigen, dass die Halbleiterfertigung mit jeder Technologiegeneration teils neuen und teils bereits bekannten, aber verschärften Problemen und Wechselwirkungen gegenüber steht. Diese können sich teilweise kritisch auf die Ausfallrate, die Lebensdauer, die Leistungsfähigkeit und die Leistungsaufnahme hochintegrierter Schaltungen auswirken. Aufgrund der engen Verknüpfung diverser Effekte und Mechanismen ist eine getrennte Behandlung der genannten Punkte nicht möglich, da beispielsweise die Optimierung eines Systems auf Leistungsfähigkeit immer auch Aspekte der Zuverlässigkeit und Leistungsaufnahme beeinflusst. Vielmehr sind Kompromisse notwendig, um leistungsfähige und dennoch zuverlässige und energieeffiziente Systeme zu realisieren.

2.2 Zuverlässigkeit

Zuverlässigkeit und Robustheit sind essentielle Faktoren für den Betrieb hochintegrierter Schaltungen. Zuverlässigkeit ist die Eigenschaft einer Schaltung, die ihr zugewiesene Funktion in einem gegebenen Zeitintervall wie vorgesehen zu erfüllen. Dies bedeutet, dass auch im Fehlerfall die Funktionsfähigkeit dauerhaft gewährleistet sein muss. Der Fokus liegt hierbei auf einer langfristigen Perspektive, welche auf eine möglichst lange Lebensdauer einer Schaltung ausgerichtet ist. Robustheit beschreibt hingegen die Fähigkeit einer Schaltung vorübergehenden Veränderungen oder Störungen von innen und außen standzuhalten. Schwankungen der Betriebsparameter und Umgebungsbedingungen dürfen möglichst keinen Einfluss auf den Zustand des Systems haben. Daher liegt der Fokus hier auf einer kurzfristigen, den aktuellen Betrieb betreffenden Perspektive.

Der Schwerpunkt dieser Arbeit liegt auf zuverlässigkeitbezogenen Problemstellungen und zielt damit auf Maßnahmen mit langfristiger Wirkung sowie eine Erhöhung der Lebensdauer. Aus diesem Grund bleibt im Folgenden Robustheit weitgehend unberücksichtigt. Dabei darf allerdings nicht vergessen werden, dass auch die Beseitigung oder Eindämmung robustheitsmindernder Effekte zu einem zuverlässigeren Betrieb und einer längeren Lebensdauer hochintegrierter Schaltungen beitragen kann.

2.2.1 Grundlagen und Definitionen

Prinzipiell muss bei zuverlässigkeitorientierten Betrachtungen zwischen den Fehlerursachen („Faults“), den dadurch auftretenden Fehlern („Errors“) und dem daraus resultierenden Fehlverhalten beziehungsweise Ausfall einer Schaltung („Failure“) unterschieden werden [HBB⁺11]. Das Fehlverhalten einer Schaltung ist als ihr Unvermögen zu interpretieren, die ihr zugedachte Aufgabe beziehungsweise Funktionsweise für eine definierte Dauer und unter festgelegten Bedingungen korrekt auszuführen [Iee90]. Diese Unterscheidung erlaubt eine Klassifizierung der Ursachen für das Fehlverhalten hochintegrierter Schaltungen, wie sie in Abbildung 2.4 [C.11] dargestellt ist und in Abschnitt 2.2.2 eingehender erläutert wird.

Ausfallrate Die Ausfallrate $\lambda(t)$ ist ein geeigneter Parameter, um Bewertungen bezüglich der Zuverlässigkeit hochintegrierter Schaltungen durchzuführen. Sie gibt an, wie viele Komponenten einer Schaltung in einem vorgegebenen Zeitintervall durchschnittlich ein fehlerhaftes Verhalten aufweisen oder ausfallen. Beispielsweise bedeutet eine Fehlerrate von monatlich 25 %, dass ein Viertel aller Komponenten in einem Monat ausfällt oder, dass eine bestimmte Komponente im Verlauf eines Monats mit einer Wahrscheinlichkeit von 25 % ausfällt. Diese Rate hängt neben der Zeit von einer Vielzahl weiterer Parameter ab.

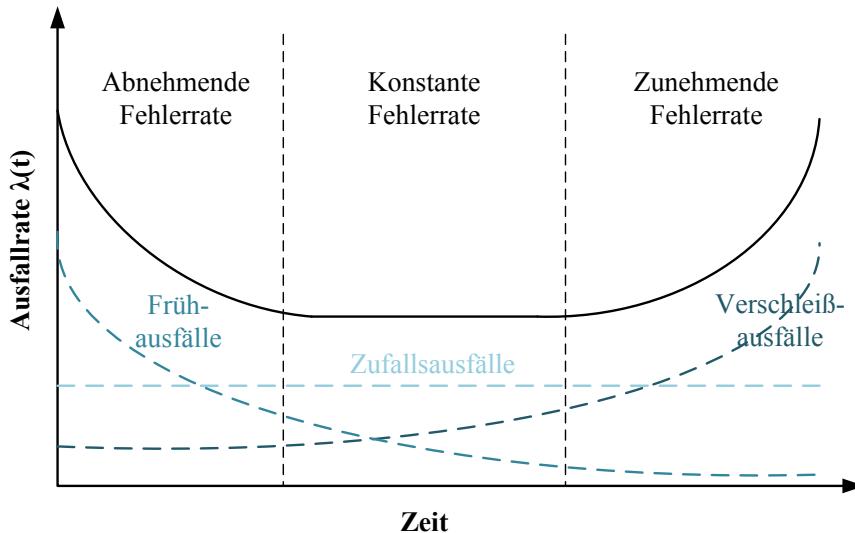


Abbildung 2.3: Lebenszyklus hochintegrierter Schaltungen: Darstellung des Zusammenhangs zwischen Lebensdauer und Ausfallrate in der Badewannenkurve

Erfahrungsgemäß folgt der Verlauf der Ausfallrate während der Lebensdauer eines Systems oder einer Schaltung der sogenannten Badewannenkurve [Iee90, KKW03]. Wie in Abbildung 2.3 zu erkennen ist, resultiert der badewannenförmige Funktionsverlauf der Ausfallrate aus den drei Einzelkomponenten der Frühausfälle, der Zufallsausfälle und der durch Verschleiß verursachten Ausfälle. Frühausfälle treten vorrangig in der ersten Phase der Lebensdauer einer Schaltung auf. Diese Phase zeichnet sich durch eine relativ hohe, aber rasch sinkende Ausfallrate aus und endet idealerweise vor der endgültigen Inbetriebnahme der Schaltung. Die Ausfallursachen in dieser Phase sind in der Regel Mängel im Entwurfs- und Produktionsprozess, sodass beispielsweise statische Fehler entstehen (siehe Abschnitt 2.2.2). In der zweiten Phase liegt die Ausfallrate auf konstant niedrigem Niveau. Da nur sporadisch auftretende Zufallsausfälle für fehlerhaftes Schaltungsverhalten sorgen, ist dies die Phase der effektiven Lebensdauer einer Schaltung. Mit dem Voranschreiten der Zeit findet der Übergang in die Phase der Spätausfälle statt. Während dieser sorgen Alterungs- und Verschleißerscheinungen für einen erneuten Anstieg der Ausfallrate, da viele Komponenten das Ende ihrer Lebensdauer erreicht haben.

Zuverlässigkeit Mithilfe der Ausfallrate $\lambda(t)$ lässt sich die Zuverlässigkeit $R(t)$ einer Schaltung nach Gleichung 2.2 berechnen. $R(t)$ entspricht damit dem Gegenteil der Definition von Fehlverhalten und gibt die Wahrscheinlichkeit an, dass eine Komponente zum Zeitpunkt t ordnungsgemäß funktioniert [Jed11]. Der Verlauf von $R(t)$ wird mithilfe der Weibull-Verteilung beschrieben. Durch die Anpassung des Parameters β können alle drei Phasen der Lebensdauer einer Schaltung repräsentiert werden [Jed11].

$$R(t) = e^{-\lambda \cdot t^\beta} \begin{cases} \beta < 1 & \text{für Frühausfälle} \\ \beta = 1 & \text{für Zufallsausfälle} \\ \beta > 1 & \text{für Verschleißausfälle} \end{cases} \quad (2.2)$$

Mean Time To Failure Die Mean Time To Failure (MTTF) gibt die erwartete mittlere Lebensdauer einer Schaltung an [Jed11]. Sie geht damit den zu $R(t)$ entgegengesetzten Weg und gibt die durchschnittliche Zeit an, nach der eine Schaltung ein fehlerhaftes Verhalten aufweist oder ausfällt. Die MTTF errechnet sich nach Gleichung 2.3 aus dem Integral der Zuverlässigkeit $R(t)$, wobei eine konstante Fehlerrate λ mit $\beta = 1$ für die Weibull-Verteilung angenommen wird. Dementsprechend ergibt sich die MTTF als Kehrwert der Ausfallrate. Dies gilt aufgrund der Annahme $\beta = 1$ allerdings nur für die Phase der effektiven Lebensdauer.

$$MTTF = \int_0^\infty R(t) dt = \frac{1}{\lambda} \quad \text{für } \beta = 1 \quad (2.3)$$

2.2.2 Ausfallursachen

Abbildung 2.4 stellt eine Kategorisierung der Ursachen für das Fehlverhalten hochintegrierter Schaltungen dar. Das aus einem Fehler resultierende Fehlverhalten ist nicht dargestellt. Die zu diesem Fehlverhalten führenden „Errors“ und die jeweiligen Ursachen lassen sich aber nach dem abgebildeten Schema klassifizieren. So unterscheidet man zunächst zwischen Fehlern, die zu einem dauerhaften oder vorübergehenden Fehlverhalten führen. Die Kategorie der temporären Fehler wird an dieser Stelle nur kurz angerissen, da der Fokus dieser Arbeit auf der Verhinderung beziehungsweise dem Hinauszögern dauerhafter Fehler liegt.

Temporäre Fehler Vorübergehend bestehende Fehler treten während des Betriebs auf und betreffen nicht die physische Schaltung als solche, sondern die verarbeiteten oder gespeicherten Daten. Die Ursachen hierfür lassen sich in die zwei Gruppen der transienten und der Timing-Fehler einteilen.

Transiente Fehler werden durch Strahlung oder Umwelteinflüsse hervorgerufen. Für strahlungsbedingte Fehler sind Partikeleinschläge und Elektromagnetische Interferenzen (EMI) [RFFM03] verantwortlich. Partikeleinschläge werden beispielsweise von α -Partikeln oder Neutronen [HZL⁺07, Nic05] verursacht. Beide Effekte rufen Veränderungen der in einzelnen Schaltungsteilen gespeicherten Ladungen hervor [MKSZ05, Bau05]. Durch Umwelteinflüsse hervorgerufene negative Effekte ergeben sich beispielsweise aus der Umgebungstemperatur oder Schwankungen der Energieversorgung [TB10].

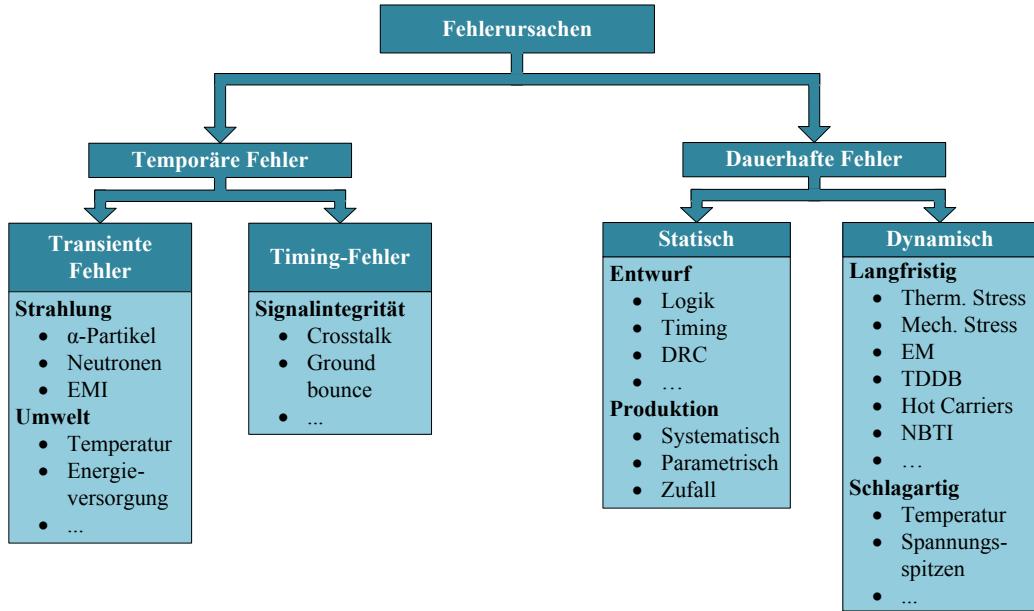


Abbildung 2.4: Klassifikation der Ursachen für das Fehlverhalten hochintegrierter Schaltungen nach Dauer, Zeitpunkt des Auftretens und Ursprung [C.11]

Timing-Fehler resultieren aus die Signalintegrität betreffenden Phänomenen. Hierzu zählen Effekte wie Crosstalk [GWWT12] oder Ground Bounce [CGB97], welche zu einer gegenseitigen Beeinflussung benachbarter Leitungen oder unerlaubten Übergängen in Zustandsmaschinen führen können.

Dauerhafte Fehler Permanent bestehende Fehler treten während des Betriebs einer hochintegrierten Schaltung auf und bleiben für die restliche Lebensdauer bestehen. Dabei wird zwischen statischen und dynamischen Fehlern unterschieden.

Statische Fehler beruhen auf Entwurfs- oder Produktionsfehlern, sind bereits zu Beginn der Lebensdauer einer Schaltung vorhanden und treten erst bei Inbetriebnahme in Erscheinung. Auf diese Art der Fehler kann während des Betriebs kein Einfluss genommen werden. Zu der Gruppe der Entwurfsfehler zählen unter anderem Logikfehler während der Implementierung, Fehler während der Timing-Analyse oder Fehler in automatisierten Entwurfsschritten wie Design Rule Checking (DRC) oder Layout Versus Schematic (LVS). Produktionsfehler treten in Form von Zufallsfehlern auf, wie sie beispielsweise durch Werkstofffehler oder Verunreinigungen des Wafers hervorgerufen werden. Weitere Produktionsfehler sind parametrischer oder systematischer Natur. Parametrische Fehler sind das Resultat von Parameterschwankungen während der Fertigung, welche beispielsweise Fehler bei Oxidier-, Ätz- oder Dotierungsvorgängen zur Folge haben. Die korrekte Funktionalität ist dabei zwar prinzipiell gegeben, jedoch können vorgegebene Spezifikationen bezüglich der Leistungsfähigkeit oder der Leistungsaufnahme nicht ein-

gehalten werden. Systematische Fehler ergeben sich aus Unzulänglichkeiten während des Fertigungsprozesses und sind damit reproduzierbar. Hierzu gehören beispielsweise fehlerhafte Fotolithografiemasken oder generell Fehler in einzelnen Prozessschritten.

Die Gruppe der dynamischen Fehler ist für diese Arbeit von größerem Interesse, da diese Mechanismen enthält, welche während des Schaltungsbetriebs zu Alterungs- und Verschleißerscheinungen beitragen. Diese Mechanismen lassen sich weiter bezüglich der Form ihres Auftretens in schlagartige und langfristige Effekte einteilen. Phänomene schlagartigen Fehlverhaltens haben ihren Ursprung meist in unsachgemäßer Behandlung oder dem Betreiben außerhalb vorgegebener Betriebsparameter. Hierzu zählen unter anderem zu hohe Betriebs- und Umgebungstemperaturen oder auch Spannungsspitzen, wie sie beispielsweise durch elektrostatische Entladungen entstehen können. Auch mechanische Belastungen wie Stöße oder Erschütterungen können eine Schaltung beschädigen. Generell führen solche Erscheinungen zu einer sofortigen Beschädigung oder Zerstörung der Schaltung, können aber durch relativ einfache Maßnahmen vermieden werden. Von größerer Bedeutung sind dagegen langfristige Phänomene, welche aufgrund von Alterung und Verschleiß über teils sehr lange Zeiträume zu einer Verschlechterung des Schaltungsverhaltens oder zu einer Beschädigung der Schaltung führen. Da diese Fehlerkategorie von zentralem Interesse für diese Arbeit ist, wird sie im Folgenden gesondert aufgeführt.

Langfristige Effekte Thermischer und mechanischer Stress zählen zu den wichtigsten langfristigen Ursachen für Alterungs- und Verschleißerscheinungen. Thermischer Stress entsteht durch lokale oder globale Temperaturspitzen sowie häufige Temperaturschwankungen und starke Temperaturgefälle zwischen einzelnen Bereichen einer Schaltung. Mechanischer Stress hat seinen Ursprung beispielsweise in mechanischen Spannungen, welche auf die Komponenten einer Schaltung einwirken.

Des Weiteren trägt der Effekt der Elektromigration (EM) [Lie06] maßgeblich zu Verschleißerscheinungen bei. EM beschreibt den Materialtransport in Leiterbahnen infolge von Ionenbewegungen. Dieser Materialtransport wird durch den Stromfluss innerhalb der Leiterbahnen verursacht und führt zur Bildung von Hohlräumen (Voids) und Ablagerungen (Hillocks). Die Folge sind Leitungsunterbrechungen beziehungsweise Kurzschlüsse.

Time Dependent Dielectric Breakdown (TDDB) [Sta02, VSE⁺⁰⁰] hingegen beeinträchtigt die Transistorfunktionsfähigkeit infolge hoher Feldstärken. TDDB definiert sich dabei als die langsam voranschreitende Zerstörung der als Isolator genutzten Oxid-Schicht zwischen Gate und Kanal eines Transistors, in welcher zunehmend Tunnelströme auftreten. Die Zerstörung dieser Schicht hat letzten Endes einen Kurzschluss zwischen Gate und Kanal zur Folge, sodass der Transistor funktionsunfähig ist.

Einen ähnlichen Effekt hat das Phänomen der heißen Ladungsträger (Hot Carriers) [CSGS85] beziehungsweise der Hot Carrier Injection (HCI). Heiße Ladungsträger sind aufgrund ihrer hohen kinetischen Energie in der Lage, in das Gate eines Transistors einzudringen und somit Leckströme zu verursachen. Oder sie verbleiben im Gate-Oxid und bilden Charge Traps, welche die Schaltgeschwindigkeit und weitere Transistorkennwerte beeinträchtigen. Ihr hohes Maß an kinetischer Energie erhalten die heißen Ladungsträger durch die Beschleunigung innerhalb eines starken elektrischen Feldes.

Der Effekt der Negative Bias Temperature Instability (NBTI) [SB03] betrifft p-Kanal-MOSFETs (Metal Oxide Semiconductor Field Effect Transistor) und wird durch negative Gate-Spannungen in Kombination mit hohen Temperaturen hervorgerufen. NBTI verursacht Charge Traps an der Grenzfläche zwischen Gate-Oxid und Transistorkanal sowie eine Verschiebung der Schwellspannung und ähnelt damit bezüglich der Auswirkungen stark dem Effekt der HCI. Weitere zu Verschleiß und Alterung führende Mechanismen sind beispielsweise Surface Inversion und Stress Migration [Jed11], welche zu Leckströmen und Materialtransport beziehungsweise -schwund führen.

Ein Großteil der für Alterung und Verschleiß einer Schaltung verantwortlichen Mechanismen ist temperaturabhängig und wird durch hohe Temperaturen beschleunigt. Deshalb konzentriert sich diese Arbeit im Folgenden auf die Eindämmung und Abschwächung dieser Effekte, da auf diese Weise Verschleiß- und Ausfallerscheinungen am effektivsten entgegen gewirkt werden kann. Eine detailliertere Betrachtung der Temperaturabhängigkeit der betreffenden Mechanismen findet im Folgenden statt.

2.2.3 Einfluss der Temperatur

In Abschnitt 2.2.2 wurden unter anderem verschiedene physikalische Mechanismen vorgestellt, welche langfristig die Lebensdauer hochintegrierter Schaltungen beeinträchtigen. Aufgrund des signifikanten Einflusses der Temperatur werden in diesem Abschnitt nun die wichtigsten solcher Mechanismen näher beleuchtet, um aufzuzeigen, inwiefern die Reduzierung der Temperatur zu einer Erhöhung der Zuverlässigkeit beitragen kann.

Lebensdauer und Beschleunigung von Alterungerscheinungen Generell gilt das Arrhenius-Modell [Jed11] als Ausgangspunkt für die Beschreibung der Temperaturabhängigkeit von Ausfallursachen. Mithilfe von Gleichung 2.4 [Jed11] ist es möglich, eine grobe Aussage über die temperaturbedingte Time To Failure (TTF), also die Dauer bis zum Eintreten von Alterungs- und Verschleißerscheinungen sowie Ausfällen, zu treffen. Dabei ist E_A die Aktivierungsenergie des betreffenden, zum Ausfall führenden Mechanismus, k_{Boltz} ist die Boltzmann-Konstante ($8,62 \cdot 10^{-5}$ eV/K) und T ist die Temperatur. Aus diesem Zusammenhang ergibt sich eine Beschleunigung von Ausfällen, welche mit steigender Temperatur exponentiell zunimmt.

Es ist zudem möglich, einen Faktor A_T für die Beschleunigung [Jed11] eines konkreten Mechanismus bei einer Veränderung der Temperatur zu bestimmen (siehe Gleichung 2.5). Dies setzt eine konstante Ausfallrate (siehe Abschnitt 2.2.1) sowie bekannte Werte für E_A und T voraus. A_T repräsentiert damit das Verhältnis zwischen den unterschiedlichen, für verschiedene Bedingungen vorherrschenden Ausfallraten beziehungsweise TTFs. λ_1 und λ_2 sind die jeweiligen Ausfallraten für die Temperaturen T_1 und T_2 .

$$TTF \propto e^{\frac{E_A}{k_{Boltz} \cdot T}} \quad (2.4)$$

$$A_T = \frac{\lambda_{T_1}}{\lambda_{T_2}} \propto e^{\frac{-E_A}{k_{Boltz}} \cdot (1/T_1 - 1/T_2)} \quad (2.5)$$

Elektromigration Der bereits vorgestellte Mechanismus der EM basiert auf Materialtransport infolge von Kollisionen zwischen den Ladungsträgern eines durch einen Leiter fließenden Stromes und den Ionen des Leitermaterials. Der Effekt verstärkt sich mit steigenden Stromdichten aufgrund sinkender Leiterquerschnitte, welche im Zuge von Strukturverkleinerungen unvermeidbar sind. Dieser Sachverhalt ist vereinfacht für einen quadratischen Leiterquerschnitt in Gleichung 2.6 dargestellt. \vec{J} ist die Stromdichte, a ist die Seitenlänge des Leiterquerschnitts und I ist der Strom. Wird beispielsweise die Seitenlänge a halbiert, vervierfacht sich die Stromdichte. Dieser theoretisch quadratische Einfluss von Strukturverkleinerungen auf die Stromdichte reduziert sich in der Realität, da die Leiterhöhe in der Regel nicht in gleichem Maße verkleinert wird. Dies ändert jedoch nichts an der Relevanz dieses Effektes. Darüber hinaus steigt die Stromdichte an Stellen im Leiter an, an denen sich Hohlräume bilden, da dort der Leiterquerschnitt im Laufe der Zeit abnimmt.

Die TTF_{EM} beschreibt die Dauer bis zum Auftreten von Ausfällen durch EM und berechnet sich nach Gleichung 2.7, welche dem Black'schen Gesetz folgt [Cle97]. J und J_{krit} bezeichnen die aktuelle sowie die für EM kritische Stromdichte. N ist der Stromdichteexponent, E_A ist die Aktivierungsenergie für EM und T ist die Temperatur. Für Kupferleitungen gelten zum Beispiel $1,1 < N < 2$ und $E_A = 0,85$ bis $0,95$ eV [Jed11]. Neben der Stromdichte beeinflusst somit auch die Temperatur die EM. Die temperaturbedingte Beschleunigung A_T ist mithilfe der Gleichungen 2.5 und 2.7 in Abbildung 2.5 für Temperaturerhöhungen von 1°C , 5°C und 10°C ausgehend von 30°C , 40°C und 50°C dargestellt. Wie zu erkennen ist, wird A_T sehr schnell größer, sodass sich bei einer Temperaturerhöhung von 10°C bereits eine Beschleunigung um das 3-fache. Dies ist als eine Verkürzung der TTF auf ein Drittel des Ausgangswertes zu interpretieren. Eine Erhöhung der Ausgangstemperatur auf 50°C führt hingegen zu einer Reduzierung der Beschleunigung auf das 2,6-fache, was mit einer an diesem Punkt bereits niedrigeren TTF zu erklären ist, wodurch sich das Verhältnis zu TTFs höherer Temperaturen verkleinert.

Die Voraussetzung für solche Berechnungen ist die Annahme, dass alle Parameter außer der Temperatur konstant bleiben. Für E_A wird ein Wert von 0,9 eV angenommen [Jed11].

$$\vec{J} = \frac{\vec{I}}{\vec{A}} \propto \frac{1}{a^2} \quad (2.6)$$

$$TTF_{EM} \propto (J - J_{krit})^{-N} \cdot e^{\frac{E_A}{k\text{Boltz}\cdot T}} \quad (2.7)$$

Time Dependent Dielectric Breakdown Ein ähnlicher Effekt ist für TDDB zu beobachten. Die durch TDDB verursachte, langfristige Zerstörung der Oxid-Schicht zwischen Gate und Kanal eines Transistors beruht auf hohen Feldstärken. Diese verursachen minimale Tunnelströme, welche sich aufgrund des Kreislaufs aus zunehmenden Strömen und sich erhöhenden Temperaturen im Laufe der Zeit schädigend auf die Oxid-Schicht auswirken und diese zerstören.

Die Dauer bis zum Auftreten von TDDB-bedingten Ausfällen TTF_{TDDB} wird mit Hilfe von Gleichung 2.8 berechnet [Jed11]. Dabei ist γ ein temperaturabhängiger Feldbeschleunigungsparameter, E_{ox} ist das von außen einwirkende elektrische Feld, E_A ist die Aktivierungsenergie für TDDB und T ist die vorherrschende Temperatur. Für γ und E_A gelten Werte von 2,5 bis 3,5 cm/MV beziehungsweise 0,6 bis 0,9 eV [Jed11]. E_{ox} ergibt sich als der Quotient aus Spannung und Oxid-Dicke in MV/cm. Neben der anliegenden Spannung, der Oxid-Dicke sowie dem elektrischen Feld hat auch die Temperatur einen Einfluss auf TDDB-bedingte Erscheinungen. In Abbildung 2.5 ist ein zu EM identisches Verhalten erkennbar. Der absolute Unterschied zwischen den einzelnen Werten für A_T lässt sich mit der niedrigeren Aktivierungsenergie von 0,75 eV [Jed11] erklären. Diese ist neben der Temperatur der einzige Faktor, der bei der Berechnung von A_T einen Einfluss hat. Alle weiteren Parameter werden als konstant angenommen und γ wird mit 3 cm/MV festgelegt [Jed11]. Die stark ausgeprägte Beschleunigung der zur Zerstörung der Oxid-Schicht führenden Abläufe verdeutlicht auch hier den signifikanten Einfluss der Temperatur.

$$TTF_{TDDB} \propto e^{-\gamma \cdot E_{ox}} \cdot e^{\frac{E_A}{k\text{Boltz}\cdot T}} \quad (2.8)$$

Hot Carrier Injection HCI beschreibt das Phänomen, durch welches heiße Ladungsträger bis in das Gate eines Transistors vordringen oder im Gate-Oxid verbleiben. Das dafür nötige hohe Maß an kinetischer Energie erhalten die heißen Ladungsträger durch die Beschleunigung innerhalb eines starken elektrischen Feldes. HCI hat sowohl Leckströme als auch eine langfristige Verschlechterung der Transistorkennwerte zur Folge. Letzteres wird auch als Device Degradation bezeichnet und umfasst unter anderem Änderungen der Schaltgeschwindigkeit sowie der Schwell- und Sperrspannung.

Die TTF_{HCl} für n-Kanal-Transistoren ist in Gleichung 2.9 beschrieben [Jed11]. I_{sub} ist der Strom ins Substrat des Transistors, N ist eine konstante ganze Zahl von 2 bis 4, E_A ist die Aktivierungsenergie für HCl mit Werten von -0,2 bis +0,4 eV und T ist die Temperatur. Für p-Kanal-Transistoren existieren zwei weitere HCl-Modelle, welchen je nach Kanallänge des Transistors unterschiedliche TTF-Gleichungen zugrunde liegen. Abbildung 2.5 zeigt den Temperatureinfluss auf HCl für n-Kanal-Transistoren, welcher mit steigenden Temperaturen abnimmt. Die Ursache hierfür ist einerseits die für das Beispiel mit -0,15 eV negativ festgelegte Aktivierungsenergie (N wurde mit 3 angenommen). Des Weiteren sorgen eine reduzierte Ladungsträgermobilität, eine reduzierte Anzahl heißer Ladungsträger sowie die Temperaturabhängigkeit einzelner Transistorparameter (z. B. Schwellspannung (siehe Gleichung 2.11) oder Fermi-Potential) für eine reduzierte Empfindlichkeit gegenüber HCl [BGR⁺97]. HCl ist somit ein Mechanismus, dessen Auswirkungen durch steigende Temperaturen abgeschwächt werden. Dies geschieht allerdings auf Kosten der temperaturbedingten Drift einzelner Transistorkennwerte und sorgt somit lediglich für eine andere Form der Device Degradation.

$$TTF_{HCl,n-Kanal} \propto (I_{sub})^{-N} \cdot e^{\frac{E_A}{k_B T}} \quad (2.9)$$

Negative Bias Temperature Instability NBTI ähnelt in seinen Auswirkungen stark HCl, da es hier in ähnlicher Art und Weise zu Device Degradation sowie einer dauerhaften Verringerung der Ladungsträgermobilität kommt. Im Gegensatz zu HCl betrifft NBTI allerdings nur p-Kanal-Transistoren und entsteht durch negative Gate-Spannungen und vergleichsweise hohe Temperaturen von über 100 °C, wie sie üblicherweise während der Burn-In-Phase oder dem Betrieb von Hochleistungs-ICs auftreten [SB03].

Gleichung 2.10 beschreibt die TTF für NBTI-induzierte Effekte [Jed11]. E_A ist die Aktivierungsenergie für NBTI, T ist die Temperatur, V_G ist die Gate-Spannung, α ist der Exponent für die Gate-Spannung und n ist der Zeit-Exponent. Abbildung 2.5 zeigt, dass NBTI mit steigender Temperatur beschleunigt wird. Diese Beschleunigung ist aber zumindest in Bereichen unter 100 °C mit einem Faktor von 1,1 relativ niedrig. Für das Beispiel wurden E_A mit 0,1 eV [CLKK06], α mit 3,5 und n mit 0,25 festgelegt [Jed11]. Trotz des scheinbar geringen Einflusses gewinnt NBTI mit jeder neuen Technologiegeneration an Bedeutung, da jeder Skalierungsschritt die Anfälligkeit gegenüber NBTI erhöht und somit beispielsweise eine Verschiebung der Schwellspannung begünstigt. Der Grund hierfür liegt in der Dichte der Charge Traps, welche mit der durch Skalierungsprozesse dünner werdenden Dicke des Gate-Oxids zunimmt [SB03].

$$TTF_{NBTI} \propto \left(e^{\frac{E_A}{k_B T}} \cdot V_G^\alpha \right)^{\frac{1}{n}} \quad (2.10)$$

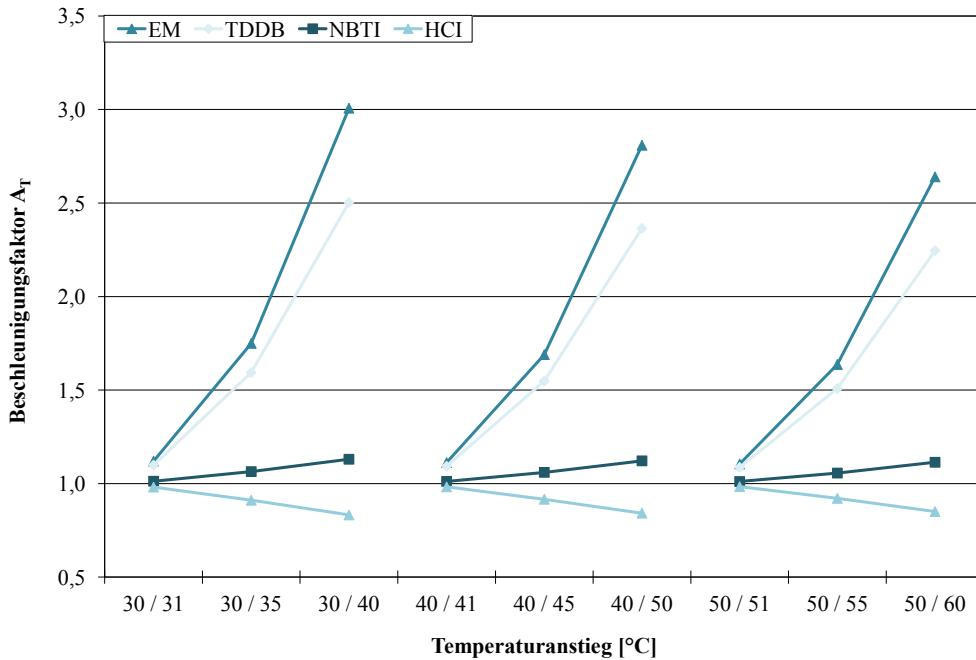


Abbildung 2.5: Einfluss eines Temperaturanstiegs um verschiedene Werte auf den Beschleunigungsfaktor A_T für EM, TDDB, HCI und NBTI

Prinzipiell lässt sich aus den gezeigten Mechanismen die Tendenz ableiten, dass eine Temperaturerhöhung signifikant zur Beschleunigung von Alterungs- und Verschleißerscheinungen beiträgt. Dies röhrt zumeist von beschleunigten elektrochemischen Reaktionen her. Das zugrunde liegende Arrhenius-Modell basiert dabei auf der Van't Hoffschen Regel, welche auch als Reaktionsgeschwindigkeit-Temperatur-Regel (RGT-Regel) bekannt ist. Diese besagt, dass sich für eine Temperaturerhöhung um 10 K die Geschwindigkeit chemischer Reaktionen verdoppelt. Übertragen auf hochintegrierte Schaltungen und Bezug nehmend auf die in Abbildung 2.5 gezeigten Ergebnisse, lässt sich damit die grobe Faustregel aufstellen, dass eine dauerhafte Temperaturerhöhung um 10 K eine Halbierung der Lebensdauer integrierter Schaltungen zur Folge hat.

Transistorkennwerte Die Drift der Schwellspannung V_{Th} ist ein Effekt, den TDDB, HCI und NBTI gemeinsam haben. Neben der Beeinflussung der Schwellspannung durch die genannten Mechanismen weist V_{Th} selbst ebenfalls eine Temperaturabhängigkeit auf. Diese Abhängigkeit wird mithilfe von Gleichung 2.11 beschrieben [GWWT12]. Die mit steigender Temperatur sinkende Schwellspannung führt einerseits zu höheren Drain-Strömen und damit zu einer höheren Schaltgeschwindigkeit. Andererseits nehmen auf diese Weise auch die Leckströme zu, welche durch Tunneleffekte und vor allem durch sogenannte Sub-Threshold-Ströme zwischen Drain und Source eines Transistors verursacht werden. Das temperaturabhängige Verhalten dieser Leckströme ist in Gleichung 2.12 be-

schrieben [DN07]. I_S ist der Sättigungsstrom, also der im Sperrbereich des Transistors zwischen Drain und Source maximal fließende Strom. V_{DS} ist die im Sperrbereich zwischen Drain und Source anliegende Spannung, welche zwischen 0 und V_{Th} liegt. q ist die Elementarladung und T die Temperatur. Wie zu erkennen ist, wächst mit steigender Temperatur der im Sperrzustand fließende Strom. Dabei bestimmt in der Praxis allerdings nicht I_S den maximalen Leckstrom, sondern die temperaturabhängige Generierung von Ladungsträgern.

$$V_{Th} = V_{Th0} + \alpha_{V_{Th}} \cdot (T - T_0) \quad \text{mit } \alpha_{V_{Th}} = -2 \dots -4 \left[\frac{mV}{K} \right] \quad (2.11)$$

$$I_L = I_S \left(1 - e^{-\left(\frac{q \cdot V_{DS}}{k_{Boltz} \cdot T} \right)} \right) \quad (2.12)$$

Darüber hinaus hat die Temperatur aber auch Einfluss auf die Leistungsfähigkeit einer Schaltung. Besonders temperaturabhängige Mechanismen, welche die Transistor-kennwerte beeinflussen und über längere Zeiträume eine Parameterdrift bewirken, haben signifikanten Einfluss auf die Leistungsfähigkeit einer hochintegrierten Schaltung. Hierzu zählen beispielsweise HCI und NBTI, welche die Schaltgeschwindigkeit der Transistoren negativ beeinflussen. Zudem ist die Mobilität der Ladungsträger temperatur-abhängig (siehe Gleichung 2.13). Diese Beziehung zwischen Temperatur und Mobilität führt zu einem mit steigender Temperatur sinkenden Drain-Strom. Die Folge ist, dass die Schaltung durch verlängerte Umladevorgänge langsamer wird.

$$\mu = \mu_0 \cdot \left(\frac{T}{T_0} \right)^{\alpha_\mu} \quad \text{mit } \alpha_\mu = -1,5 \dots -2,5 \quad (2.13)$$

2.2.4 Existierende Techniken zur Steigerung der Zuverlässigkeit

In diesem Abschnitt soll ein kurzer Abriss über existierende Ansätze und Methoden zur Erhöhung der Zuverlässigkeit hochintegrierter Schaltungen gegeben werden. Im Folgenden wird dazu eine grobe Kategorisierung des Reliability Engineering vorgenommen.

Redundanz Der Ansatz der Redundanz [Iee90] sieht eine Mehrfachauslegung einzelner Komponenten eines Systems vor. Dies kann sowohl echte Komponenten als auch Daten, Informationen und Operationen betreffen. Damit lassen sich je nach gewählter Methode räumliche, informationsbezogene und zeitliche Redundanzen erzeugen. Diese werden genutzt, um vorhandene Fehler zu unterdrücken oder deren Entstehung zu verhindern. Ein populärer Vertreter der räumlichen Redundanz ist die Triple Modular Redundancy (TMR) [Sie91]. TMR basiert auf einer dreifachen Auslegung identischer Hardware-Komponenten, welche parallel die ihr zugewiesene Funktion ausführen. Mithilfe eines Mehrheitsvotums wird ein Ergebnis erzeugt. Damit wird das möglicherweise

fehlerhafte Ergebnis einer defekten Komponente unterdrückt. Eine weitere Möglichkeit TMR zu nutzen, ist die zusätzlichen Komponenten als Reserve bereit zu halten und erst dann zu aktivieren, wenn die Primärkomponente fehlerhaft ist.

Derating Derating verfolgt eine andere Strategie und verzichtet auf Redundanz. Stattdessen wird eine Komponente gewollt unterhalb ihrer maximalen Leistungsfähigkeit betrieben, um die Verlustleistung und damit die erzeugte Abwärme zu reduzieren. Ob und in welchem Umfang eine solche Drosselung stattfindet, wird durch die Auswertung verschiedener Parameter wie beispielsweise der Betriebstemperatur oder der Leistungsaufnahme entschieden. Das Ziel dabei ist, einer Beschädigung oder der Zerstörung einer Komponente vorzubeugen. Bekannte Vertreter sind das Dynamic Voltage Scaling (DVS) sowie das Dynamic Frequency Scaling (DFS) zur dynamischen Anpassung von Versorgungsspannung und Taktfrequenz. Derating wird häufig in der Luft- und Raumfahrt [ESA06] sowie im militärischen Bereich [Mil92] eingesetzt, um die Lebensdauer elektronischer Komponenten zu erhöhen.

Physics of Failure Die Strategie der Physics of Failure [GPW02] basiert auf der Identifizierung und dem Verstehen der physikalischen Prozesse und Mechanismen, welche sich hinter einem Defekt verbergen. Auf diese Weise können Fehler bereits während der Entwurfsphase durch entsprechende Maßnahmen verhindert werden. Solche Maßnahmen basieren meist auf physikalischen oder chemischen Gesetzmäßigkeiten sowie theoretischen Analysen, Simulationen und Berechnungen. Beispielsweise werden zur Verringerung des TDDB-Effekts (siehe Abschnitt 2.2.2) „high-k“-Materialien wie Hafnium für das Gate-Oxid verwendet, um Leckströme zu reduzieren.

Diese Kategorisierung ist angesichts der Vielzahl möglicher Fehlerursachen (siehe Abschnitt 2.2.2), deren Wechselwirkungen und der Komplexität hochintegrierter Systeme allerdings nicht ausreichend. In Abbildung 2.6 ist eine erweiterte Kategorisierung dargestellt, welche die bereits vorgestellten Strategien in konkreter Form umsetzt sowie um weitere Ansätze ergänzt. Diese Kategorisierung ist teilweise [C.11] entlehnt und wird im Folgenden eingehender erläutert. Die aufgeführten Maßnahmen und Strategien zur Erhöhung der Zuverlässigkeit erheben keinen Anspruch auf Vollständigkeit und stellen lediglich einen Auszug dar. Detailliertere Ausführungen zu diesem Thema sind beispielsweise [Bor05, RK09] zu entnehmen.

Fehlererkennung

Um die Zuverlässigkeit hochintegrierter Systeme durch eine gezielte Fehlerbehandlung steigern zu können, ist zunächst eine Fehlererkennung notwendig. Je nachdem in wel-

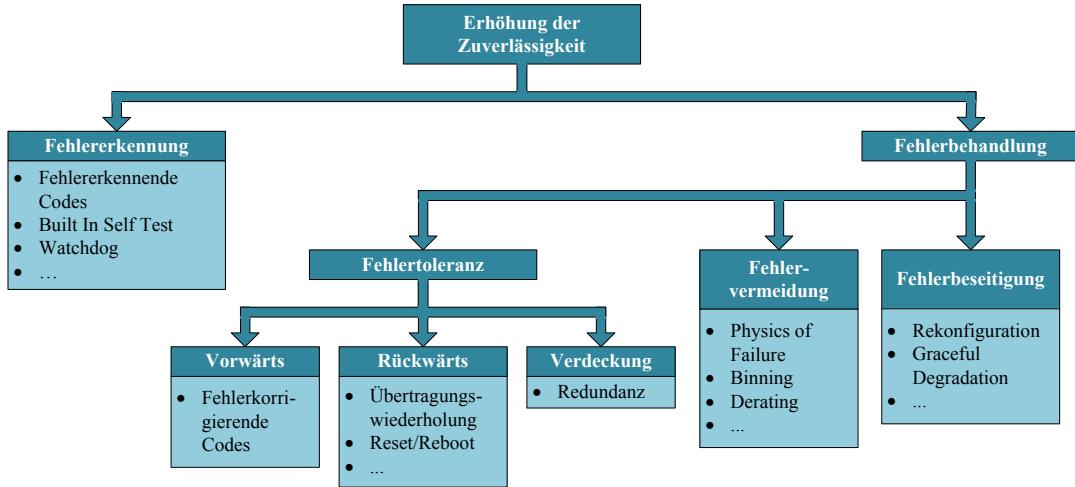


Abbildung 2.6: Kategorisierung der Strategien zur Erhöhung der Zuverlässigkeit

cher Phase (Entwurf oder Betrieb) oder auf welcher Ebene (z. B. System- oder Modulebene) auf mögliche Fehler geprüft werden soll, eignen sich verschiedene Ansätze. Fehlererkennende Datencodierungen [FKCC06, DT07] werden bei der Datenübertragung und -speicherung eingesetzt, um beispielsweise transiente Fehler (siehe Abschnitt 2.2.2) zu erkennen. Damit eignen sich solche Verfahren nur für den Betrieb. Bekannte Vertreter sind die zyklische Redundanzprüfung (Cyclic Redundancy Check (CRC)), Hash-Funktionen sowie die Verwendung von Paritätsbits und XOR-Prüfsummen.

Auf Modul- und Systemebene eignen sich Built In Self Tests (BIST) [McC85] für eine Fehlererkennung sowohl während des Entwurfs als auch während des Betriebs. Zu diesem Zweck besitzt eine Schaltung zusätzliche Logik, welche Testmuster erzeugt und mit den vorgegebenen korrekten Resultaten vergleicht. Damit lassen sich vor allem statische Fehler lokalisieren. BIST-Varianten sind Analog-, Mixed-Signal- und Logik-BIST sowie der Boundary Scan Test [RTZ98].

Eine weitere Möglichkeit der Fehlererkennung ist der Einsatz sogenannter Watchdogs [Lam12]. Dies sind Systemkomponenten, welche den Betrieb anderer Komponenten überwachen und die Erkennung eines möglichen Fehlers oder Problems melden, so dass eine Fehlerbehandlung stattfinden kann. Watchdogs können als Software und Hardware ausgelegt sein und dementsprechend entweder die Funktion anderer Hardware-Komponenten beziehungsweise Software-Module überwachen. Generell lässt sich auch die Überwachung verschiedener Systemparameter (z. B. Übertragungsdauer oder Daten- und Bitfehlerrate) zur Erkennung von Fehlern nutzen, da drastische Schwankungen solcher Parameter auf ein Problem hindeuten können.

Fehlerbehandlung

Grundlegend existieren drei verschiedene Ansätze auf erkannte Fehler zu reagieren. Diese unterscheiden sich in ihren konkreten Realisierungen teils stark und beziehen sich oftmals auf die eingangs vorgestellte Einteilung in Redundanz, Derating und Physics of Failure.

Fehlertoleranz Die Strategie der Fehlertoleranz hat das Ziel, auch im Falle eines Fehlers in einer oder mehreren Komponenten für einen möglichst reibungslosen und fehlerfreien Betrieb zu sorgen. Hierfür stehen drei Möglichkeiten zur Verfügung.

Die Methode der Vorwärts-Fehlertoleranz stützt sich auf die Kompensation existierender Fehler durch fehlerkorrigierende Codes [FKCC06, DT07] und eignet sich daher für eine Fehlerbehandlung bei Datenübertragungen und -speicherungen. Dabei werden den Daten Rekonstruktionsinformationen hinzugefügt, sodass im Falle eines Fehlers eine Wiederherstellung erfolgen kann. Zu solchen Codes zählen die Trellis-Kodierung, der Reed-Solomon-Code, Faltungscodes und mehrdimensionale Paritätscodes aber auch Kodierungen zur Vermeidung von Crosstalk [PGF⁺06]. Der Nachteil fehlerkorrigierender Codes ist der verringerte Datendurchsatz aufgrund der zusätzlichen Korrekturdaten.

Für eine Rückwärts-Fehlertoleranz sind unter anderem simple Übertragungswiederholungen im Falle eines detektierten Fehlers möglich. Ein Beispiel hierfür ist das Automatic Repeat Request (ARQ), wobei der Empfänger einer Datenübertragung deren Fehlerfreiheit entweder bestätigt oder eine Übertragungswiederholung erbittet. Diese Methode ist bei permanenten Fehlern allerdings wirkungslos. Weitere Möglichkeiten sind Resets des Systems in vorherige, als fehlerfrei bekannte Zustände sowie komplette Reboots einzelner Komponenten oder des gesamten Systems. Ist ein erkannter Fehler nicht korrigierbar, so muss er maskiert werden.

Bei der Fehlerverdeckung werden hauptsächlich die eingangs vorgestellten Redundanzverfahren wie TMR eingesetzt, um einen Fehler mithilfe von Mehrheitsentscheidungen zu eliminieren. Die redundant ausgelegten Komponenten können auf verschiedene Art und Weise genutzt werden. Einerseits kann die Ausfallsicherheit durch verschiedene Ansätze wie aktive, passive, Standby- und N+1-Redundanz beeinflusst werden [Sie91]. Andererseits ist auch ein paralleler energiesparender Betrieb der redundanten Komponenten möglich [SCS⁺10].

Fehlervermeidung Die Strategie der Fehlervermeidung hat hingegen das Ziel, Fehlern durch entsprechende Maßnahmen vorzubeugen. Hierzu zählt beispielsweise der Ansatz der Physics of Failure. Das Wissen um die Ursachen für Erscheinungen wie TDDB, EM oder Crosstalk sowie eventuelle Wechselwirkungen wird eingesetzt, um bereits während des Entwurfs geeignete Gegenmaßnahmen einzuleiten. Um zum Beispiel EM abzu-

schwächen, wird Wire Widening betrieben, also die Vergrößerung des Leiterquerschnitts zur Senkung der Stromdichte. Wire Spreading und Wire Filling sind weitere Maßnahmen. Ersteres sieht die Vergrößerung des Abstandes benachbarter Leitungen zur Reduzierung von Koppelkapazitäten vor, während Letzteres das Ausfüllen des Raumes benachbarter Leitungen mit abschirmenden Materialien zur Reduzierung von Crosstalk zum Ziel hat. Mit Via Duplication, Via Shifting und Via Widening existieren ähnliche Ansätze auch für die Verbindungen zwischen den einzelnen Ebenen eines Chip-Layouts.

Auch das bereits vorgestellte Verfahren des Derating lässt sich zur Vermeidung von Fehlern durch die dynamische Anpassung der Leistungsfähigkeit einer Schaltung einsetzen. Die Methode des Binnings [SLN98] ähnelt der des Deratings. Binning wird jedoch vor der Inbetriebnahme einer Schaltung eingesetzt, um durch Analysen und Tests eventuelle Fertigungsfehler oder Unzulänglichkeiten zu identifizieren. Je nach Art und Auswirkung des Fehlerbildes werden verschiedene Exemplare einer digitalen Schaltung unterschiedlich klassifiziert. Anhand dieser statischen Klassifikation werden die Exemplare mit einer entsprechend angepassten Leistungsfähigkeit betrieben.

Fehlerbeseitigung Eine weitere Strategie der Fehlerbehandlung ist die Fehlerbeseitigung. Bekannte Defekte oder Fehler sollen dabei bereits während des Entwurfs aus dem aktiven Teil eines Systems ausgeschlossen werden.

Dies wird unter anderem durch statische Rekonfigurationen erreicht, wobei ein als fehlerhaft erkannter Teil eines Systems durch sogenannte Anti-Fuses dauerhaft vom Rest des Systems getrennt wird. Aufgrund der Beschaffenheit der Anti-Fuses ist dieser Vorgang nur einmal möglich und irreversibel. In [CK95] wird ein Ansatz vorgestellt, bei dem während der Fertigung mithilfe von Laser Fuses fehlerhafte Komponenten vom System getrennt und an deren Stelle Reservekomponenten eingebunden werden. Auf diese Weise können zum Beispiel Mehrkernprozessoren, welche aufgrund von Fertigungsfehlern defekte oder fehlerhafte Kerne aufweisen, durch eine Abschottung dieser mit reduzierter Kernanzahl weiterbetrieben werden.

Eine weitere Möglichkeit der Fehlerbeseitigung ist die Graceful Degradation. Dabei wird während des Betriebs gezielt auf einen Fehler oder einen Defekt reagiert, indem der angebotene Funktionsumfang oder auch dessen Qualität schrittweise reduziert wird. Zu diesem Zweck werden die betroffenen Komponenten entweder gänzlich deaktiviert oder in einen sicheren Betriebszustand versetzt. Die Gesamtfunktionalität des Systems soll dabei soweit wie möglich erhalten bleiben. Der Prozess der Graceful Degradation ist zumindest theoretisch reversibel.

Die in diesem Abschnitt aufgeführten Methoden stellen nur einen Auszug der Möglichkeiten zur Steigerung der Zuverlässigkeit dar, doch sie geben einen guten Überblick über

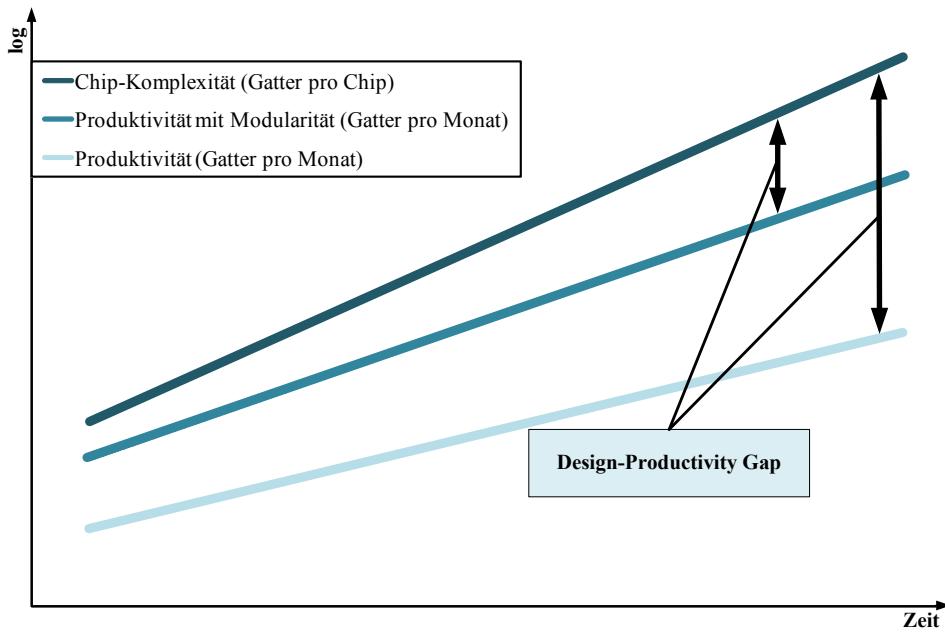


Abbildung 2.7: Design-Productivity Gap: Diskrepanz zwischen technologisch möglicher Chip-Komplexität und tatsächlicher Entwurfseffizienz [ITR13a]

die verschiedenen strategischen Ansätze. Besonders vor dem Hintergrund der Verzögerung temperaturbedingter Alterungs- und Verschleißerscheinungen ist die Strategie der Fehlervermeidung ein sinnvoller Ausgangspunkt. Das Wissen um die Ursachen und Zusammenhänge der zu Verschleiß und Alterung führenden physikalischen Mechanismen gepaart mit dynamischen Ansätzen wie Derating scheint dabei am vielversprechendsten zu sein und wird in Kapitel 4 erneut aufgegriffen.

2.3 Kommunikationsstrukturen nanoelektronischer Systeme

Wie bereits in Abschnitt 2.1 ausgeführt wurde, sorgen neue Fertigungstechnologien und damit immer kleinere Strukturgrößen für steigende Integrationsdichten (siehe Abbildung 2.2). Dies eröffnet die Möglichkeit immer komplexere Schaltungen auf einem einzelnen Chip zu integrieren, angefangen bei einfachen Funktionen bis hin zu mehreren vollständigen Rechenkernen. Die Kehrseite der somit immer komplexeren und leistungsfähigeren Chips ist der wachsende Design-Productivity-Gap (DPG) (siehe Abbildung 2.7). Dieser beschreibt die Schere zwischen technologisch möglicher Chip-Komplexität und der demgegenüber stehenden tatsächlichen Entwurfseffizienz [ITR12, ITR13b].

Eine vielversprechende Möglichkeit diese Diskrepanz zu überwinden, stellt das Prinzip des modularen Chip-Entwurfs dar. Mit dessen Hilfe können hochkomplexe On-Chip-

Systeme in kleinere, weniger komplexe und funktional abgeschlossene Komponenten eingeteilt werden [Ope13, Nio13, Arm13]. Auf diese Weise kann der DPG reduziert werden [ITR13b], indem während des Entwurfs mit abstrakten Modulen gearbeitet wird. Diese verbergen die eigentliche Funktionalität und sorgen somit für ein gewisses Maß an Regularität. Beispiele für solche modular gestalteten Systeme sind die Tilera TILE64 Prozessorfamilie [Til12a], der Adapteva Epiphany-IV Mikroprozessor [Ada12a] sowie der Intel Single Chip Cloud Computer (SCC) [Int12] und Intels „Tera-Scale Computing“-Programm [HBK06].

Die gewählten Beispiele sind darüber hinaus allesamt NoC-basiert. Inwiefern diese Kommunikationsarchitektur aufgrund ihrer Eigenschaften einen regulären und modularen Systementwurf unterstützt, wird im Folgenden näher erläutert. Damit einher geht ein Paradigmenwechsel von lokaler und serieller Berechnung hin zu räumlich getrennter und paralleler Berechnung mit einer strikten Trennung von Kommunikation und Verarbeitung. Der Fokus verschiebt sich von einer berechnungsorientierten zu einer kommunikationsorientierten Entwurfsmethodik [Jan03, Nur04]. Dies schlägt sich ebenfalls im Einfluss der Kommunikation auf die Gesamtleistungsaufnahme sowie die Leistungsfähigkeit eines Systems nieder.

Die On-Chip-Kommunikationsinfrastruktur, welche die einzelnen Systemkomponenten miteinander verbindet und starken Einfluss auf viele leistungs- und zuverlässigkeitsbezogene Parameter hat, wird somit zu einem zentralen Entwurfsthema. Daher beschäftigt sich dieser Abschnitt der Arbeit mit den Organisations- und Kommunikationsstrukturen moderner mikro- und nanoelektronischer Systeme. Dazu wird zunächst in Abschnitt 2.3.1 ein Überblick über die häufigsten Architekturen erstellt. Diese werden bezüglich allgemeiner Parameter wie Leistungsfähigkeit und Komplexität gegenübergestellt und bewertet. Im Anschluss werden in Abschnitt 2.3.3 NoC-basierte Kommunikationsarchitekturen detailliert analysiert. Neben den bereits eingeführten Parametern werden diese hinsichtlich weiterer Anforderungen bewertet, welche im Rahmen dieser Arbeit relevant sind.

2.3.1 Überblick und analytischer Vergleich

Es existieren verschiedene Möglichkeiten die Komponenten eines On-Chip-Systems miteinander zu verbinden. Diese unterscheiden sich im Aufbau des Kommunikationsmediums und in der Art und Weise, wie der Zugriff auf dieses geregelt wird. Generell können Kommunikationsarchitekturen aufgrund der ihnen zugrunde liegenden Topologie in vier Gruppen eingeteilt werden [dMB06]:

- Geteiltes Medium: alle Komponenten teilen sich das Kommunikationsmedium

- Direktes Netzwerk: jede Komponente besitzt über eine Schnittstelle direkte Verbindungen zu einer Teilmenge aller Komponenten
- Indirektes Netzwerk: jede Komponente ist mit einem Router verbunden, welcher Punkt-zu-Punkt-Verbindungen zu anderen Routern besitzt
- Hybrides Netzwerk: eine Kombination der Vorherigen

Im Folgenden werden nun die drei etabliertesten Ansätze zur Verbindung der Komponenten hochintegrierter Systeme näher betrachtet. Die einzelnen zu verbindenden Systemkomponenten werden fortan aus zweierlei Gründen allgemein als Intellectual Property Core (IPC) bezeichnet. Erstens wird davon ausgegangen, dass es sich entweder um Systeme mit homogenen oder mit heterogenen Komponenten handelt [dMB06]. Dies bedeutet, dass ein System entweder aus identischen IPCs mit gleicher Funktionalität besteht (z. B. Manycore-Systeme), welche parallel verschiedene Aufgaben bearbeiten. Oder ein System besteht aus IPCs verschiedener Funktionalität, wobei nur wenige Aufgaben parallel bearbeitet werden (z. B. Mobilgeräte). Zweitens wird für die folgenden Betrachtungen angenommen, dass aufgrund der Modularität keine genaue Kenntnis über die eigentliche interne Funktionalität eines IPC vorliegt. Diese ist allerdings auch nur bedingt notwendig, da der Fokus ohnehin auf der Kommunikation zwischen den IPCs sowie der Auswirkung der verschiedenen Topologien auf folgende Parameter liegt:

- Leistungsfähigkeit: Datendurchsatz und Latenz der Datenübertragung
- Komplexität der Kommunikationsarchitektur
- Flächenbedarf, Leistungsverbrauch
- Skalierbarkeit und Wiederverwendbarkeit

Punkt-zu-Punkt-Verbindung Die Topologie der Punkt-zu-Punkt-Verbindung stellt die einfachste Art dar, IPCs miteinander zu vernetzen und gehört zur Kategorie der direkten Netzwerke. Diese Art der Topologie eignet sich besonders für spezialisierte Systeme, da nur tatsächlich miteinander kommunizierende IPCs über dedizierte Verbindungen zu einander verfügen (siehe Abbildung 2.8(a)). Die Verbindung zwischen zwei IPCs sowie deren Schnittstellen sind daher auf den spezifischen Datenverkehr zugeschnitten und vergleichsweise simpel gestaltet. Dies hat den Vorteil, dass keine zusätzliche Logik für Arbitrierung, Routing oder Flusskontrolle erforderlich ist.

Auf Punkt-zu-Punkt-Verbindungen basierende Systeme zeichnen sich durch eine sehr hohe Leistungsfähigkeit aus [LCOM08]. Datendurchsatz und Latenz werden lediglich durch die Auslegung der Verbindung und der Schnittstellen beschränkt. Trotz der einfachen Kommunikationsstruktur weist dieser Ansatz bezüglich Flächenbedarf und

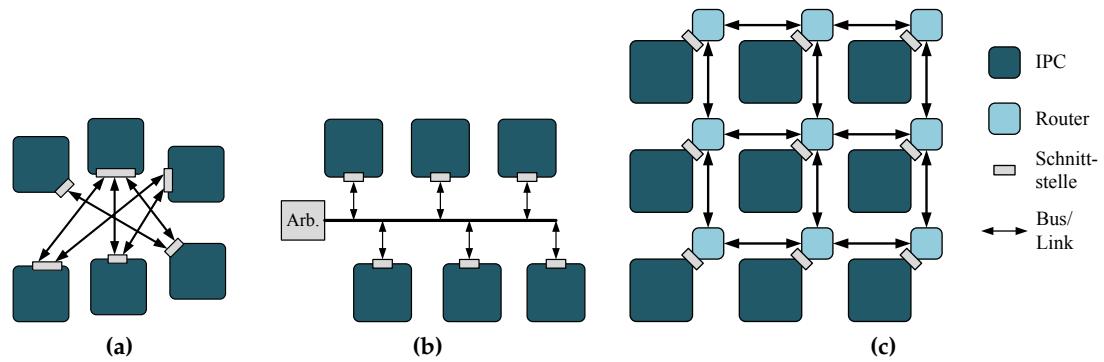


Abbildung 2.8: Kommunikationsarchitekturen: (a) Punkt-zu-Punkt-Verbindung, (b) gemeinsam genutzter Bus, (c) NoC am Beispiel eines 3×3 Gitternetzes

Leistungsverbrauch schlechte Werte auf. Die Ursache hierfür liegt in der mangelhaften Skalierbarkeit beim Hinzufügen weiterer IPCs. So steigen die Kosten für Fläche, globale Leitungslänge und Leistungsverbrauch [LCOM08] für jeden zusätzlichen IPC unverhältnismäßig stark an, da neben der zusätzlichen Komponente auch mehrere neue Schnittstellen und Verbindungen notwendig sind. Die Leistungsfähigkeit nimmt dagegen durch zusätzliche Verbindungen nur in sehr geringem Maße zu.

Aufgrund der Spezialisierung Punkt-zu-Punkt-basierter Systeme weisen diese zudem einen geringen Grad der Wiederverwendbarkeit auf, da die Kommunikationsstrukturen an die jeweiligen Kommunikationspaare und deren Anforderungen angepasst sind. Aus diesen Gründen eignen sich Punkt-zu-Punkt-Verbindungen vorrangig für weniger komplexe Systeme oder für den Fall, dass die Anforderungen an die Kommunikationsarchitektur bereits während der Entwurfsphase bekannt und unveränderlich sind.

Busbasierte Systeme Busbasierte On-Chip-Systeme sind der Kategorie der Netzwerke mit geteiltem Medium zuzuordnen und stellen die am weitesten verbreitete Kommunikationsarchitektur dar [Cor99, RMI04]. Im einfachsten Fall verbindet ein gemeinsamer Bus alle IPCs eines Systems miteinander (siehe Abbildung 2.8(b)). Eine Kommunikation zwischen den IPCs findet ausschließlich über diesen Bus statt, wobei nur ein IPC gleichzeitig senden kann, aber mehrere IPCs parallel empfangen können. Da alle IPCs auf dasselbe Medium zugreifen und über dasselbe Protokoll kommunizieren, müssen die zugehörigen Schnittstellen für eine konsistente Anbindung an den Bus sorgen. Damit sind die Schnittstellen deutlich komplexer als bei Punkt-zu-Punkt-Verbindungen, da häufig heterogene IPCs verbunden werden müssen.

Um eine reibungslose Kommunikation zu gewährleisten, muss jeder IPC zunächst über den Arbiter (siehe Abbildung 2.8(b)) Zugriff auf den Bus beantragen. Ist der Zugriff gewährt, können die Daten übertragen und der Bus wieder freigegeben werden.

Die Leistungsfähigkeit busbasierter Systeme ist stark begrenzt [LCOM08] und hängt von der Anzahl angebundener IPCs ab, da der maximal mögliche Datendurchsatz unter allen Teilnehmern aufgeteilt wird. Die Latenz einer einzelnen Datenübertragung hängt hauptsächlich von der Anzahl der Zugriffsanfragen und der Arbitrierungsstrategie [SRK11] ab. Einerseits nimmt die Arbitrierung selbst Zeit in Anspruch. Andererseits entscheidet die Strategie darüber welcher IPC wann Zugriff auf den Bus erhält.

Bezüglich Flächenbedarf, globaler Leitungslänge und Leistungsverbrauch schneiden busbasierte Systeme mit einer wachsenden Anzahl zu verbindender IPCs deutlich besser ab als Punkt-zu-Punkt-Topologien [LCOM08]. Die zunächst dominanten, durch den Arbiter und die komplexeren Schnittstellen hervorgerufenen, Mehrkosten relativieren sich schnell. So kommen mit jeder weiteren Komponente lediglich eine neue Verbindung und eine neue Schnittstelle hinzu. Zusätzlich steigt die Komplexität des Arbiters minimal. Daher sind busbasierte Systeme bezüglich der Kosten sehr viel besser skalierbar als Punkt-zu-Punkt-Topologien. Dies trifft allerdings nicht auf die Leistungsfähigkeit zu. Zum einen reduziert jede weitere Komponente die theoretisch mögliche Zugriffszeit jedes IPCs. Zum anderen müssen zusätzlich häufiger Arbitrierungen durchgeführt werden, welche zudem mehr Zeit in Anspruch nehmen.

Aufgrund des einfachen und für alle Komponenten identischen Anbindungskonzeptes weisen busbasierte Systeme einen hohen Grad der Wiederverwendbarkeit auf. Die Bus-Topologie eignet sich durch ihre Eigenschaften bedeutend besser für die Vernetzung der Komponenten komplexer Systeme als Punkt-zu-Punkt-Verbindungen. Dies betrifft vor allem die kostengünstigere Anbindung. Da sich allerdings die Leistungsfähigkeit busbasierter Systeme mit einer steigenden Anzahl von IPCs verschlechtert, sind diese nur bedingt für komplexe Systeme mit hohen Leistungsanforderungen geeignet.

Networks-on-Chip (NoCs) NoCs unterscheiden sich grundlegend von busbasierten und Punkt-zu-Punkt-Topologien und zeichnen sich insbesondere durch eine sehr gute Skalierbarkeit aus. Sie sind der Kategorie der indirekten Netzwerke zuzuordnen, wobei eine Vielzahl konkreter Topologien möglich ist. Diese werden in Abschnitt 2.3.3 näher erläutert. Die einfachste, aufgrund ihrer Regularität am weitesten verbreitete Form [SKH08] ist das 2D-Gitternetz, welches in Abbildung 2.8(c) dargestellt ist.

Prinzipiell greifen NoC-basierte Systeme die Idee der paketbasierten Kommunikation zwischen den Teilnehmern eines verteilten Netzwerks (z. B. das Internet) auf und übertragen diese auf On-Chip-Systeme. Die jeder Komponente angeschlossene Schnittstelle sorgt für eine konsistente Anbindung aller IPCs an das NoC, da von einer heterogenen Natur des Systems ausgegangen werden muss. Anstelle eines geteilten Mediums sind die Systemkomponenten allerdings über ein Netzwerk miteinander vernetzter Router verbunden. Aufgrund der modularen Auslegung sowohl der IPCs als auch des Kom-

Tabelle 2.1: Qualitative Bewertung und Vergleich Punkt-zu-Punkt-basierter, busbasierter und NoC-basierter Kommunikationsarchitekturen [C.11]

	Punkt-zu-Punkt	Busbasiert	Network-on-Chip
Komplexität der Netzwerkelemente	niedrig	gehoben zzgl. Arbiter	hoch
Datendurchsatz	hoch, garantiert	begrenzt	hoch
Latenz	niedrig, determinist.	niedrig nach Arbitrierung	schwankt mit Auslastung und Distanz
Skalierbarkeit: Leistungsfähigkeit	begrenzt	nein	ja
Skalierbarkeit: Kosten	nein	ja	ja
Wiederverwendbarkeit	ungeeignet	eingeschränkt	geeignet

munikationsnetzwerks sind einerseits mehrere parallele Datenströme möglich. Andererseits werden dadurch auch teils aufwendige Verfahren für Arbitrierung, Flusskontrolle und Routing erforderlich, welche in den Routern durchgeführt werden müssen.

Der Datendurchsatz NoC-basierter Systeme ist vergleichsweise hoch [LCOM08], da mehrere Kommunikationen parallel stattfinden können. Allerdings sind sowohl Latenz als auch Datendurchsatz stark von der Art des auftretenden Verkehrs abhängig. Die Latenz hängt zudem von der Netzauslastung und der Entfernung zwischen Sender und Empfänger ab. Auch die eingesetzten Algorithmen für Routing, Arbitrierung und Flusskontrolle spielen dabei eine wichtige Rolle. Die erhöhte Komplexität der Netzwerkelemente wirkt sich zwar negativ auf Leistungsverbrauch und Flächenbedarf aus [LCOM08], wird aber durch die sehr gute Skalierbarkeit bezüglich Leistungsfähigkeit und auch kostenbezogener Parameter aufgewogen. So bedeutet eine zusätzliche Komponente einen neuen Router und mindestens eine neue Verbindung. Diese tragen wiederum zur Erhöhung des Durchsatzes [LCOM08] und zur Entlastung des Netzwerks bei und beeinflussen somit auch die Latenz positiv. Zusätzliche Komponenten beeinflussen Flächenbedarf, globale Leitungslänge sowie den Leistungsverbrauch stärker als dies bei busbasierten Systemen der Fall ist. Die entstehenden Kosten sind jedoch gut kalkulierbar, da die bestehende Infrastruktur nicht modifiziert oder erweitert werden muss.

Die Netzwerkelemente weisen durch die streng modulare Gestaltung einen hohen Grad der Wiederverwendbarkeit auf, da aufgrund der konsistenten Anbindung der IPCs

kaum Anpassungen nötig, aber sehr gezielt möglich sind. Das Modularitätsprinzip trägt zudem zur Abschwächung des DPG bei (siehe Abbildung 2.7). Besonders die gute Skalierbarkeit und der hohe Datendurchsatz NoC-basierter Kommunikationsarchitekturen machen deutlich, dass diese sich hervorragend für die Vernetzung komplexer On-Chip-Systeme eignen. Nachteilig sind die schlechte Vorhersagbarkeit der Latenz, die Komplexität der Netzwerkelemente und die Gefahr der Überdimensionierung bei Systemen mit geringen Leistungsanforderungen. Eine qualitative Bewertung von NoCs sowie der zuvor vorgestellten Kommunikationsarchitekturen ist Tabelle 2.1 zu entnehmen.

2.3.2 Grundlagen NoC-basierter Systeme

Im Folgenden werden grundlegende NoC-spezifische Aspekte, aber auch allgemeine netzwerktheoretische Begrifflichkeiten näher erläutert, welche für diese Arbeit relevant sind. Der Einsatz des Prinzips der paketbasierten Kommunikation in NoC-basierten Systemen erfordert eine entsprechende Aufbereitung der zwischen den einzelnen Systemkomponenten ausgetauschten Informationen (siehe Abbildung 2.9). Die kleinste in einem Takt auf einem Übertragungskanal des NoC übertragbare Dateneinheit ist die Physical Control Unit (Phit). Die Größe eines Phits entspricht der Datenbreite des Übertragungskanals und wird in Bits angegeben. Üblich sind beispielsweise 32, 64, 128 oder 256 Bit.

Um einen reibungslosen Ablauf zu gewährleisten und die Konkurrenz der verschiedenen Datenströme um die begrenzten Ressourcen des Netzwerks zu regeln, werden Mechanismen zur Flusskontrolle eingesetzt. Diese werden auf Ebene der Flow Control Unit (Flit) durchgeführt. Ein Flit besteht dabei aus mindestens einem Phit. Detaillierte Ausführungen zum Thema Flusskontrolle sind beispielsweise [dMB06] zu entnehmen. Mehrere Flits bilden Pakete, welche letztendlich die Nachrichten repräsentieren, die zwischen den Komponenten ausgetauscht werden. Die Unterteilung in Pakete ist aufgrund von Beschränkungen oder Spezifikationen des Netzwerks nötig. Diese würden andernfalls unter Umständen durch die Größe einer Nachricht verletzt. Die Paket- und Nachrichten-Header enthalten Steuerinformationen wie beispielsweise die Zieladresse oder die Länge der Nachricht beziehungsweise des Pakets. Vereinfachend oder aus Gründen der Flusskontrolle wird oft eine einheitliche Größe für Phit und Flit definiert, sodass nur noch der Begriff Flit Verwendung findet. Dies gilt auch für den weiteren Verlauf dieser Arbeit.

NoC-Komponenten

NoCs bestehen aus den Komponenten Router, Netzwerkschnittstelle und Link. Diese drei Bestandteile werden im Folgenden näher erläutert.

Router Der Router, oder auch Switch, ist ein aktives Netzwerkelement und ist für das Weiterleiten eintreffender Pakete verantwortlich. Der Aufbau eines Routers ist in Abbil-

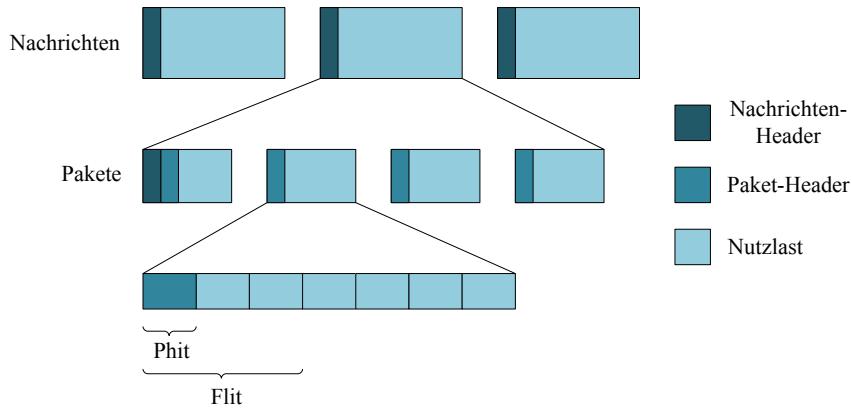


Abbildung 2.9: Aufbau einer Nachricht

dung 2.10 dargestellt. Den als Port bezeichneten Router-Komponenten werden entsprechend ihrer Lage die vier Himmelsrichtungen zugewiesen. Die Ports dienen als Bindeglied zum Übertragungskanal in die jeweilige Richtung. Der an den Router angeschlossene IPC erhält ebenfalls einen Port. Jeder Port setzt sich wiederum aus einem Eingangs-, einem Ausgangs- und einem Puffermodul zusammen. Der als First-In-First-Out-Speicher (FIFO) realisierte Puffer dient dem Zwischenspeichern am Eingangsmodul eintreffender Flits, falls diese nicht sofort weitergeleitet werden können. Die Wahl der Größe dieses Speichers hat großen Einfluss auf den Platzbedarf und die Leistungsaufnahme eines Routers [C.11]. Das Eingangsmodul eines jeden Ports ist über die Verdrahtungsmatrix mit den Ausgangsmodulen aller anderen Ports verbunden.

In den Ein- und Ausgangsmodulen werden je nach Implementierung unterschiedliche Funktionen realisiert. Üblicherweise findet im Eingangsmodul das Routing statt. Dabei wird auf Basis vordefinierter Regeln entschieden, an welchen Port das aktuelle Paket weitergeleitet werden soll. Eine Klassifikation verschiedener Routing-Verfahren findet im Anschluss statt. Im Ausgangsmodul hingegen findet die Arbitrierung statt. Dabei wird entschieden, von welchem Port das nächste Paket entgegen genommen wird. Ein Verfahren, welches eine Gleichberechtigung aller Ports gewährleistet, ist der oft eingesetzte Round-Robin-Algorithmus [DT03]. Je nach Routing-Verfahren und der Anzahl der angeschlossenen IPCs können sich Anzahl und Auslegung der Ports unterscheiden.

Netzwerkschnittstelle Die Netzwerkschnittstelle ist für die Anbindung eines IPC an das NoC verantwortlich. Da sich die Schnittstellen der verschiedenen IPCs teils deutlich unterscheiden können, aber aufgrund der Modularität trotzdem eine homogene Anbindung aller IPCs gewährleistet sein muss, ist die Netzwerkschnittstelle zweigeteilt (siehe Abbildung 2.11). Der unveränderliche Teil des Resource Independent Network Interface (RINI) ist für jeden Router identisch. Im Gegensatz dazu gestaltet sich der Teil des Re-

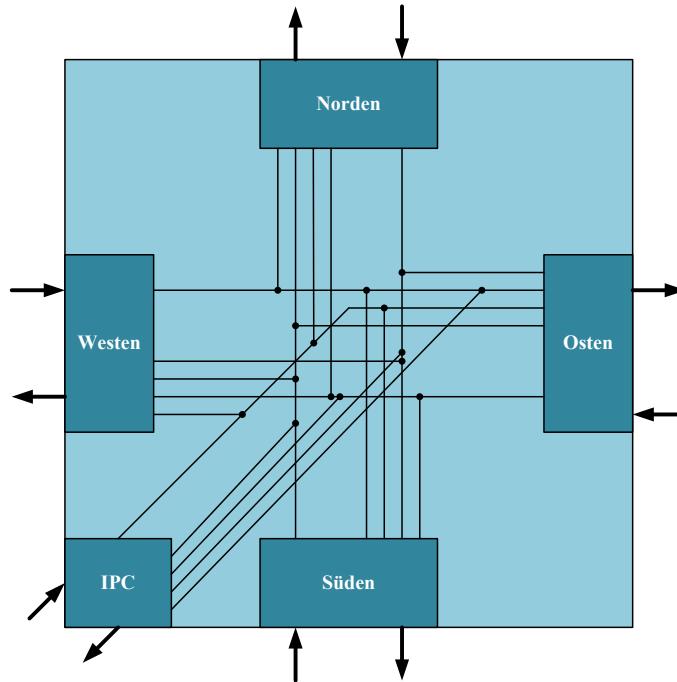


Abbildung 2.10: Aufbau eines NoC-Routers bestehend aus den Ports sowie der Verdrahtungsmatrix

source Dependent Network Interface (RDNI) je nach angebundenem IPC unterschiedlich. Diese Aufteilung sorgt für eine homogene und doch flexible Anbindung sämtlicher IPCs an das Netzwerk, da das RDNI entsprechend den Anforderungen des IPC modifiziert werden kann und das RINI stets identisch ist.

Übertragungskanal Der Übertragungskanal, oder auch Link, ist ein passives Netzwerkelement und stellt die Verbindung zwischen zwei Routern her. Er besteht aus zwei unidirektionalen Punkt-zu-Punkt-Bussen und ist aus physikalischer Sicht lediglich eine Menge von Leitungen auf dem Chip. Diese Leitungen unterteilen sich in Daten- und Steuerleitungen. Letztere dienen ausschließlich der Übertragung von Kontroll- und Statussignalen. Hierzu zählen beispielsweise Signale für die Flusskontrolle. Die Datenbreite eines Flits entspricht der Anzahl der Datenleitungen, sodass je Datenübertragung ein Flit übertragen wird.

Switching

Die Methode des Switching bestimmt die Technik, mit der Informationen im Netzwerk weitergeleitet und in den einzelnen Routern zwischengespeichert werden. Zur Festlegung des Switchings gehören die Wahl der Switching-Technik und die Wahl der Einheit eines einzelnen Datentransfers, also der Größe eines Flits.

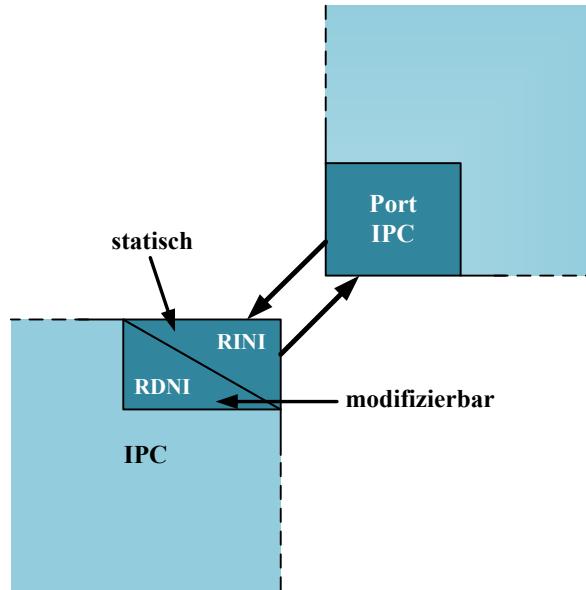


Abbildung 2.11: Netzwerkschnittstelle zwischen NoC-Router und angebundenem IPC

Packet-Switching Packet-Switching unterteilt eine Nachricht in einzelne, voneinander unabhängige Pakete, welche auf unterschiedlichen Pfaden vom Sender zum Empfänger übertragen werden können.

Das Store-And-Forward Switching (SAFS) [dMB06] ist die simpelste Methode des Packet-Switching. Dabei wird ein Paket erst dann zum nächsten Router übertragen, wenn der Puffer des Eingangssports am Empfänger genügend freien Speicher für das gesamte Paket bereithält. Hierdurch entsteht eine gewisse Verzögerung, da umgekehrt das Paket am Empfänger vollständig vorliegen muss, bevor dessen Header-Inhalt abgerufen wird und es weitergeleitet werden kann.

Das Virtual-Cut-Through Switching (VCTS) [dMB06] leitet hingegen das Header-Flit eines Pakets sofort weiter. Voraussetzung ist, dass der nachfolgende Empfänger über ausreichend freien Speicher verfügt, um das komplette Paket zu speichern. Ist dies der Fall, werden alle Daten-Flits ohne Verzögerung ebenfalls weitergeleitet. Ist nicht genügend Speicher verfügbar, so wird das gesamte Paket zwischengespeichert.

Das Wormhole-Switching (WHS) [dMB06] reduziert den bereitzustellenden Speicher auf ein Flit je Port. Bei dieser Methode werden die Flits eines Pakets mithilfe eines Handshake-Verfahrens weitergeleitet, sobald der nachfolgende Router das Flit zwischen-speichern kann. Andernfalls wird gewartet bis genügend Speicher zur Verfügung steht. Dieses Verfahren wird von fast allen NoCs genutzt, da die Größe der Eingangspuffer drastisch reduziert wird. Dies wiegt auch den Nachteil auf, dass bei höheren Verkehrs-dichten vorübergehende Blockierungen der Übertragungskanäle wahrscheinlicher sind, da Flits aufgrund der beschränkten Puffergröße nicht sofort weitergeleitet werden kön-

nen. Packet-Switching eignet sich eher für kurze Nachrichten, da durch die Einteilung in Pakete ein gewisser Mehraufwand entsteht.

Circuit-Switching Das Circuit-Switching [dMB06] teilt eine Nachricht nicht in Pakete ein, sondern überträgt diese im Ganzen als Datenstrom. Die Übertragung unterteilt sich in die drei Phasen des Verbindungsaufbaus, der eigentlichen Übertragung und des Verbindungsabbaus. Während des Verbindungsaufbaus wird der Pfad vom Sender zum Empfänger mittels des Header-Flits reserviert. Ist das Ziel erreicht, wird dem Sender die Pfadreservierung bestätigt, sodass mit der verzögerungsfreien Übertragung begonnen werden kann. Während dieser Zeit ist der Pfad für andere Nachrichten blockiert. Nach Beendigung der Übertragung wird die Pfadreservierung aufgehoben. Diese Methode eignet sich besonders für lange Nachrichten, da die verzögerungsfreie Übertragung trotz des Verbindungsaufbaus einen Zeitvorteil gegenüber Packet-Switching erzielt.

Routing

Die Routing-Strategie bestimmt unabhängig vom Switching den Pfad eines Pakets durch das Netzwerk und hat damit maßgeblichen Einfluss auf die Leistungsfähigkeit eines NoC-basierten Systems und die Latenz einzelner Übertragungen. Grundsätzlich lassen sich alle Routing-Algorithmen nach der Routing-Entscheidung, ihrer Adaptivität und der resultierenden Pfadlänge kategorisieren [dMB06]. Diese Variante der Kategorisierung ist in Abbildung 2.12 dargestellt.

Adaptivität Die Eigenschaft der Adaptivität beschreibt die Fähigkeit eines Routing-Algorithmus, Informationen über die im Netzwerk vorherrschenden Bedingungen in die Wahl des Routing-Pfades einzubeziehen. So ändert sich im Laufe der Zeit mit dem Zustand des Netzwerks auch die Pfadwahl, um kritische Netzbereiche zu umgehen. Üblicherweise beziehen sich die Netzwerkbedingungen auf die Auslastung und Stausituationen in Teilen des Netzes [DT03]. Im Rahmen dieser Arbeit sind diesbezüglich allerdings ausschließlich Ausfälle von Routern und Übertragungskanälen von Interesse.

Deterministische Algorithmen [DT03] liefern für ein bestimmtes Sender-Empfänger-Paar unabhängig vom Netzwerkstatus immer denselben Pfad. Sie sind daher besonders geeignet, wenn die zu erwartenden Verkehrsmuster im Voraus bekannt und statisch sind. Die geringe Komplexität solcher Algorithmen reduziert die Hardware-Kosten.

Teilweise adaptive Routing-Algorithmen beziehen hingegen Informationen bezüglich des Netzwerkstatus in die Routing-Entscheidung ein. Sie beschränken jedoch die Anzahl der wählbaren Pfade durch Vorgaben wie die maximale Pfadlänge und werden daher oft als minimale adaptive Routing-Algorithmen [DT03] bezeichnet.

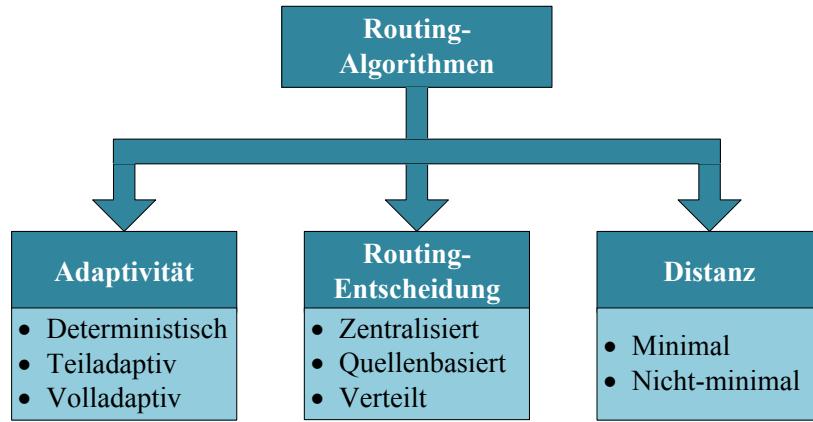


Abbildung 2.12: Klassifikation der Routing-Algorithmen

Diese Einschränkung wird für die volladaptiven Routing-Algorithmen [DT03] aufgehoben, sodass die Pfadwahl vollkommen flexibel stattfinden kann. Solche Algorithmen eignen sich für zeitlich veränderliche Verkehrsmuster. Der erhöhte Aufwand adaptiver Algorithmen wirkt sich allerdings nachteilig auf die Router-Komplexität und somit auf die Hardware-Kosten aus.

Routing-Entscheidung Die Kategorie der Routing-Entscheidung beschreibt den Ort sowie die Art und Weise, mit der das Routing vollzogen wird.

Bei verteiltem Routing [FRD08] trifft jeder Router entlang des Routing-Pfades eine eigene Entscheidung, in welche Richtung ein Paket weitergeleitet werden soll. Hierfür muss jeder Paket-Header die Zieladresse des Pakets enthalten. Diese wird von jedem Router ausgelesen, sodass mithilfe einer Lookup-Tabelle oder einer fest implementierten Routing-Logik der zu wählende Ausgang des Routers ermittelt werden kann.

Quellenbasiertes Routing [GDR05, DT03] verzichtet dagegen auf die Eintragung der Zieladresse im Paket-Header. Stattdessen wird der gesamte Routing-Pfad einmalig am Sender berechnet. Entsprechende Routing-Anweisungen werden in den Paket-Header eingetragen, wobei jeder Router entlang des Pfades die für ihn relevanten Informationen extrahiert. Der Eintrag im Header wächst somit mit steigender Pfadlänge.

Zentralisiertes Routing [MCKW10] basiert auf einer zentralen Ansammlung aller möglichen Routing-Pfade, welche dem zugrunde liegenden Routing-Algorithmus folgen. Diese werden in die Lookup-Tabellen der Router eingetragen. Eine mithilfe einer globalen Sicht auf das Netzwerk vorgenommene Anpassung der Pfade bedingt eine Aktualisierung der Lookup-Tabellen.

Distanz Je nach Routing-Algorithmus weisen die Routing-Pfade von Sender zu Empfänger eine unterschiedliche Distanz auf. Die Pfadlänge wird meist in sogenannten

Hops, der Bewegung eines Pakets von Router zu Router, angegeben. Minimales Routing erlaubt dabei lediglich Routing-Pfade, welche bezüglich der Länge dem kürzesten möglichen Pfad zwischen Sender und Empfänger entsprechen. Umgekehrt wird bei nicht-minimalem Routing die Möglichkeit der kurzfristigen Entfernung des Pakets vom Empfänger gewährt. Damit erhöht sich zwar die Distanz, es steht aber auch eine größere Anzahl alternativer Routen zur Verfügung.

Quality of Service

Ein NoC dient der Vernetzung potentiell heterogener IPCs, welche oftmals verschiedenste Aufgaben erfüllen. Damit unterscheiden sich auch die von den IPCs generierten Verkehrsmuster. Je nach Aufgabe eines IPC ergeben sich daher auch unterschiedliche Anforderungen an die Informationsübertragung. Beispielsweise benötigen Multimediaanwendungen einen hohen Datendurchsatz, während Echtzeitanwendungen hohe Anforderungen an die Latenz stellen. Die als Quality of Service (QoS) bezeichnete Kategorisierung verschiedener Verkehrsklassen ist eng mit den eingeführten Begrifflichkeiten wie Switching, Routing und Flusskontrolle verknüpft. Sie wird aufgrund ihrer Signifikanz aber dennoch gesondert aufgeführt. Allgemein lassen sich zwei grundlegende Service-Klassen definieren [DT03], welche verschiedenen Anforderungen gerecht werden.

Best Effort Best Effort (BE) Services stellen eine minimalistische Qualitätszusicherung dar. Es wird lediglich garantiert, dass Paketübertragungen durchgeführt werden, solange dies im Rahmen der verfügbaren Ressourcen möglich ist. Garantien bezüglich der Vollständigkeit oder der Korrektheit einer Übertragung werden nicht gegeben. Übertragungen dieser Kategorie weisen in der Regel eine niedrige Priorität auf.

Guaranteed Services Guaranteed Services (GS) sichern die Vollständigkeit sowie die Korrektheit von Übertragungen zu. Zusätzlich kann ein gewisses Leistungsniveau der Übertragung garantiert werden (z. B. bezüglich Latenz, Datendurchsatz, Jitter oder Paketverlustrate). Übertragungen dieser Kategorie weisen eine höhere Priorität auf.

Die Einteilung in Service-Klassen stellt eine Aggregation einer weitaus größeren Anzahl von Verkehrsklassen dar. Diese können je nach Anforderungsprofil und Priorisierung in die BE- oder GS-Kategorie eingeordnet werden. BE-Verkehr lastet die Kommunikationskapazitäten tendenziell besser aus, da eine größere Anzahl von Übertragungen parallel stattfinden kann ohne die Restriktionen zu verletzen. GS-Verkehr sorgt dagegen beispielsweise durch Latenzbeschränkungen für ein gewisses Maß an Vorhersagbarkeit und Echtzeitfähigkeit.

2.3.3 NoC-Topologien

Wie bereits in Abschnitt 2.3.1 ausgeführt wurde, eignen sich NoCs aufgrund ihrer hervorragenden Skalierbarkeit und Modularität bestens für die Vernetzung einer großen Zahl von On-Chip-Komponenten. Die Voraussetzungen sind dabei entsprechend hohe Anforderungen an die Leistungsfähigkeit und die Flexibilität der Kommunikationsarchitektur. Andernfalls sind die gegenüber einer busbasierten Realisierung anfallenden Mehrkosten nicht gerechtfertigt. Neben der bereits erwähnten Gitternetz-Topologie existiert eine Vielzahl unterschiedlichster NoC-Implementierungen, welche sich sowohl durch die Topologie als auch die angewandten Verfahren für Switching, Routing und Flusskontrolle unterscheiden [SKH08]. Im Folgenden werden die wichtigsten Topologien vorgestellt und qualitativ hinsichtlich ihrer Leistungsfähigkeit, der Kosten sowie ihrer Eigenschaften bezüglich Zuverlässigkeit und QoS verglichen und bewertet. Aus Gründen der Übersichtlichkeit ist der Detailgrad der abgebildeten Topologien gering gehalten. Ein Router samt zugehörigem IPC und der Schnittstelle werden als ein Knoten dargestellt.

Gitternetz Die bekannteste und auch am weitesten verbreitete Form der NoC-Topologien ist das n -dimensionale k -Gitter, auch Gitternetz oder Mesh genannt [KJS⁺02, WL03]. Dabei gibt n die Anzahl der Dimensionen an und k definiert die Anzahl der Knoten entlang einer Kante. Die Abbildungen 2.13(a) und 2.13(b) stellen ein 2D-Gitter und ein 3D-Gitter mit jeweils $k = 3$ dar. Neben der regulären Variante, bei der an jeden Router ein IPC angebunden ist, existieren auch Abwandlungen mit mehreren IPCs je Router [S.09]. Aufgrund ihrer Regularität skalieren Gitternetze mit wachsender NoC-Größe sehr gut. Mit jedem weiteren Knoten steigen sowohl Flächenbedarf als auch Leistungsverbrauch linear. Dies bietet den Vorteil einer gut abschätzbaren Kostensteigerung. Ein weiterer Vorteil ist ein hoher, aber von der Lastsituation abhängiger Datendurchsatz. Die reguläre Beschaffenheit ermöglicht zudem der Einsatz simpler Routing-Algorithmen. Dies führt zu einer weniger komplexen und damit sparsameren Router-Logik. Gitternetzbasierte Systeme weisen zudem einen hohen Grad der Vernetzung auf. Dieser äußert sich in der Gesamtheit aller verfügbaren Übertragungskanäle sowie im Grad der Vernetzung eines einzelnen Routers (siehe Abbildung 2.15).

Nachteilig ist die Abhängigkeit der durchschnittlichen Latenz (beziehungsweise der durchschnittlichen Anzahl benötigter Hops von Start zu Ziel) vom Durchmesser des Netzes und von der Lastsituation. Der Netzdurchmesser definiert sich als das Maximum aller minimalen Pfade zwischen allen möglichen Knotenpaaren. Beispielsweise beträgt der Durchmesser für ein 2-dimensionales 3×3 NoC 4 und für ein 4×4 NoC 6. Somit dauert mit wachsender NoC-Größe eine Übertragung zwischen zufällig ausgewählten Kommunikationspartnern im Durchschnitt länger, da sich die Entfernung zwischen diesen mit steigender Wahrscheinlichkeit erhöht.

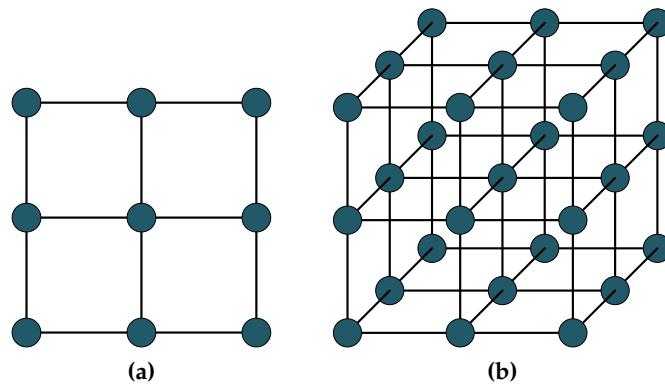


Abbildung 2.13: Gitternetz: (a) 2-dimensionales 3-Gitter, (b) 3-dimensionales 3-Gitter

Darüber hinaus besteht die Gefahr, dass simple, meist nichtadaptive Routing-Algorithmen zuverlässigkeitssbezogenen Anforderungen nicht genügen. Ein durch fehlerhafte Router oder Übertragungskanäle unzuverlässiger Kommunikationspfad kann mithilfe statischer Routing-Algorithmen in der Regel nicht umgangen werden. Daher sind komplexere und flexiblere Algorithmen nötig, um ein gewisses Maß an Zuverlässigkeit zu garantieren. Die Auswirkung von Router-Ausfällen und fehlerhaften Übertragungskanälen auf die Erreichbarkeit einzelner IPCs wird in Abschnitt 2.3.4 näher untersucht. Dabei werden Routing-Algorithmen unterschiedlicher Adaptivität berücksichtigt. In jedem Fall ist durch den vollständigen Ausfall eines Routers mindestens ein IPC gar nicht mehr erreichbar. Dies hängt davon ab, wie viele IPCs an den jeweiligen Router angeschlossen sind. Selbstverständlich ist auch der Ausfall einzelner Router-Ports möglich, sodass je nach betroffenem Port lediglich die entsprechenden Übertragungskanäle nicht mehr ansteuert werden können. Daher bieten Gitternetze ohne den Einsatz anpassungsfähiger Routing-Algorithmen lediglich einen geringen Grad an Zuverlässigkeit.

Aus diesem Grund und aufgrund der nichtdeterministischen Latenz einer Übertragung eignen sich Gitternetze daher bezüglich QoS lediglich für BE-Garantien. Dies setzt voraus, dass das NoC entweder fehlerfrei ist oder adaptive Routing-Algorithmen eingesetzt werden. Für GS sind Gitternetze aufgrund des Nichtdeterminismus ungeeignet.

Torus Eine ebenfalls weit verbreitete Topologie ist die des Torus [AEK06, CC04] (auch als k -facher n -Würfel bezeichnet). Tori stellen eine Erweiterung der Gitternetze dar, wobei die äußeren Knoten zusätzlich mithilfe globaler Übertragungskanäle verbunden werden (siehe Abbildung 2.14(a)). Dies erhöht den Grad der Vernetzung, wobei der Vorsprung gegenüber Gitternetzen allerdings mit wachsender Anzahl zu verbinder Komponenten abnimmt (siehe Abbildung 2.15). Zusätzlich wird der effektive Durchmesser des Netzes verkleinert und damit die durchschnittliche Anzahl benötigter Hops von Start zu Ziel reduziert. Die Voraussetzung ist der Einsatz adaptiver oder modifizier-

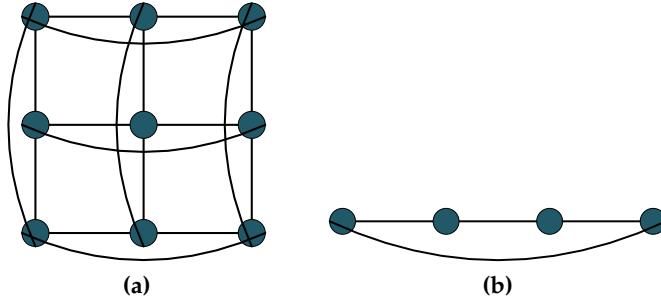


Abbildung 2.14: Torus: (a) 3-facher 2-Würfel, (b) 4-facher 1-Würfel beziehungsweise Ring

ter statischer Routing-Algorithmen, um die globalen Übertragungskanäle bei Routing-Entscheidungen berücksichtigen zu können.

Um diesen negativen Effekten entgegen zu wirken, werden die Knoten verschachtelt angeordnet, sodass ein Folded Torus entsteht [DT01]. Dies hat den Vorteil einer gleichmäßigen, wenn auch insgesamt größeren Länge der Übertragungskanäle. Eine weitere Abwandlung stellen sogenannte Hypercubes dar. Diese sind eine Kombination aus Gitternetz und Torus [CYTZ10] und zeichnen sich durch einen nochmals höheren Vernetzungsgrad aus. Die daraus resultierende Komplexität des physikalischen Chip-Layouts und der Routing-Algorithmen machen Hypercubes allerdings eher unbrauchbar für NoC-basierte Systeme. Der k -fache 1-Würfel hingegen reduziert Tori zu einer Ringstruktur mit k Knoten [STN02] (siehe Abbildung 2.14(b)). Ringe weisen mit wachsender Netzwerkgröße gegenüber Gitternetzen einen zunehmend geringeren Grad der Vernetzung auf (siehe Abbildung 2.15), da jeder Knoten immer nur auf zwei Übertragungskanäle zugreifen kann. Dies sorgt für eine vergleichsweise niedrige Zuverlässigkeit. Bereits der Ausfall eines Knotens zerstört die Ringstruktur.

Die Vorteile der regulären Torus-Topologie liegen in ihrer guten Skalierbarkeit und Regularität. Zudem weisen Tori aufgrund der zusätzlichen Übertragungskanäle einen potentiell höheren Datendurchsatz als Gitternetze auf, welcher allerdings von der Lastsituation abhängig ist. Da der höhere Grad der Vernetzung ohne zusätzliche Router erzielt wird, reduziert sich die durchschnittliche Anzahl der vom Start zum Ziel benötigten Hops. Diese ist aber nach wie vor vom Netzwerkdurchmesser abhängig. Im Vergleich zu Gitternetzen zeichnen sich Tori durch eine höhere Flexibilität aus, da mithilfe der zusätzlichen Übertragungskanäle einzelne Fehlerquellen umgangen werden können. Auch in diesem Fall setzt dies adaptive Routing-Algorithmen voraus.

Der Nachteil der Torus-Topologie sind unverhältnismäßig lange, quer über den Chip laufende Verbindungen und damit einhergehende parasitäre Effekte und hohe Signallaufzeiten. Darüber hinaus wirken sich die globalen Leitungen negativ auf den Flächenbedarf aus und erschweren das Layout, da unter Umständen andere Metalllagen genutzt werden müssen. Weiterhin steigt die Komplexität der Router am Rand und in den Ecken

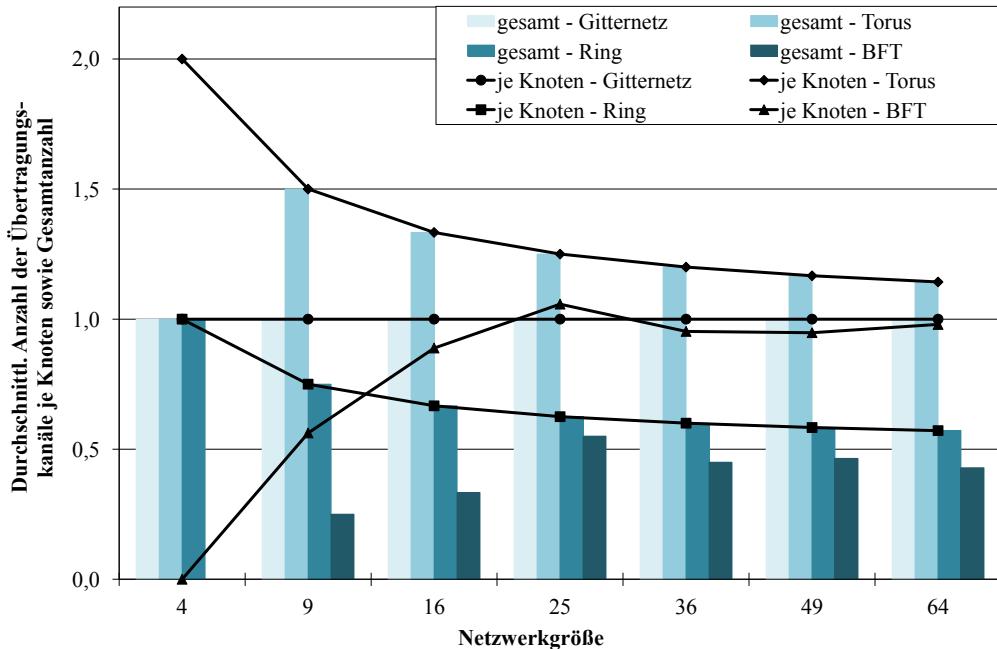


Abbildung 2.15: Grad der Vernetzung: durchschnittliche Anzahl der Übertragungskanäle je Knoten und Gesamtanzahl der Übertragungskanäle für verschiedene Topologien und Netzwerkgrößen (normiert auf Werte für die Gitternetztopologie)

des Gitters. Dort werden zusätzliche Ports benötigt, welche andernfalls eingespart werden könnten. Aufgrund längerer Leitungen sind zudem möglicherweise Repeater zur Signalverstärkung erforderlich. Die erhöhte Router-Komplexität sowie die Repeater tragen zu einem höheren Flächen- und Leistungsbedarf bei. Auch die angepassten Routing-Algorithmen wirken sich negativ aus. Diese sind nicht nur komplexer, sondern müssen auch die Freiheit von Dead- und Livelocks garantieren. Kommt es zu einem Router-Ausfall ist letztendlich auch innerhalb dieser Topologie mindestens ein IPC nicht mehr erreichbar. Auf Tori basierende Systeme sind mit entsprechend angepassten Routing-Algorithmen potentiell leistungsfähiger und zuverlässiger als gitternetzbasierte Systeme. Der Mehraufwand bleibt dabei relativ moderat.

Bezüglich QoS eignet sich auch die Torus-Topologie gut für BE-Garantien, da die globalen Übertragungskanäle eine Umgehung einzelner fehlerhafter Knoten erleichtern. GS-Garantien können auch hier nicht gegeben werden. Zwar kann aufgrund der globalen Übertragungskanäle unter Umständen die Anzahl der Hops für eine Übertragung reduziert werden. Dies trifft aber nicht unbedingt auf deren Latenz zu. Inwiefern sich der höhere Vernetzungsgrad von Torus-Topologien auf die Zuverlässigkeit auswirkt, wird in Abschnitt 2.3.4 näher untersucht.

Weitere Topologien Der k -fache n -Baum (z. B. Butterfly Fat Tree (BFT)) [PGIS03], das Oktagon [KND02] oder auch geclusterte Topologien [C.11] sind weitere Topologien. Diese eignen sich allerdings nur eingeschränkt für die Realisierung zuverlässiger On-Chip-Kommunikationsarchitekturen, da anstelle einer redundanten eine sparsame Vernetzung priorisiert wird. Auf diese Weise sollen die vorhandenen Verbindungen möglichst vollständig ausgelastet werden, um die Kommunikationswege zu verkürzen. Dabei ist der Grad der Vernetzung teils bedeutend geringer als der herkömmlicher Gitternetz- oder Torus-Strukturen (siehe Abbildung 2.15). Bei einem Ausfall eines Routers oder Übertragungskanals kann somit entsprechend weniger flexibel reagiert werden. Verglichen mit Gitternetzen weisen lediglich BFTs eine wachsende Konnektivität einzelner Knoten mit steigender Netzwerkgröße auf. Somit existieren einige zentrale und daher kritische Knoten, deren Ausfall oder Störung das Gesamtsystem stark beeinträchtigen kann. Ein Extrembeispiel ist der zentrale Verbindungsknoten der Sterntopologie. Solche Knoten sind durch ihre Lage im Netz und ihren hohen Vernetzungsgrad besonders hohen Belastungen ausgesetzt, sodass an diesen Stellen zuerst mit Verschleiß- und Alterungserscheinungen zu rechnen ist. Aus den genannten Gründen kommen diese Topologien für zuverlässigkeitsoorientierte On-Chip-Vernetzungen nicht in Frage.

Aufgrund ihrer weiten Verbreitung [SKH08] sowie guter Eigenschaften bezüglich Regularität, Skalierbarkeit und Vernetzung konzentrieren sich im Folgenden alle weiteren Untersuchungen auf 2D-Gitternetze. Besonders die vergleichsweise umfangreiche Vernetzung ist dabei bezüglich einer zuverlässigkeitsteigernden Verbindung der On-Chip Komponenten vielversprechend. Gleichzeitig lassen Regularität und Zweidimensionalität dieser Topologie geringe Entwurfskosten und einen niedrigen Entwurfsaufwand erwarten. Tori werden der Vollständigkeit halber als durch zusätzliche Übertragungskanäle modifizierte Form der Gitternetze ebenfalls betrachtet.

2.3.4 Zuverlässigkeit NoC-basierter Systeme

Dieser Abschnitt wird im Folgenden verdeutlichen, welche Auswirkungen der Ausfall einzelner Router und Übertragungskanäle auf die Zuverlässigkeit NoC-basierter Systeme hat. Zu diesem Zweck wird ein System untersucht, welches auf einem 2D-Gitternetz basiert. Weiterhin werden verschiedene minimale, deterministische und partiell adaptive Routing-Algorithmen für die Paketübertragung eingesetzt. Zu Vergleichszwecken wird ein äquivalentes torusbasiertes System betrachtet. Damit soll gezeigt werden, dass bei Ausfällen selbst relativ robuste Architekturen starke Einschränkungen hinnehmen müssen und Defekte nur bedingt kompensieren können. Zur Bewertung der Zuverlässigkeit wird die Gesamtanzahl aller übertragenen Pakete analysiert. Diese schwankt je nach Anzahl und Position der vorliegenden Fehler und ist ein Maß für die verbliebene Funktions-

Tabelle 2.2: Parameter für die Simulation der Auswirkung von Ausfällen auf gitternetz- und torusbasierte NoCs unter der Nutzung von XYR, WFR, NLR und NFR

	Parameter	Wert
Architektur	Topologie	Gitternetz, Torus
	NoC-Größe	4×4
Router	Switching	WHS
	Arbitrierung	Round-Robin
	Routing	XYR, WFR, NLR, NFR
	Flitbreite	64 Bit
Simulation	Pakete	10000
	Paketlänge	1 - 5 Flits
	Ausfallquelle	Router; Link
	Anzahl der Ausfälle	1,2; 1 ... 4

fähigkeit. Tabelle 2.2 fasst die Parameter der durchgeführten Simulationen zusammen. Vorausgesetzt alle Pakete können ungehindert das NoC durchqueren, werden während der Simulation insgesamt 10000 Pakete übertragen. Trifft ein Paket auf eine ausgefallene Komponente und kann mithilfe des eingesetzten Routing-Algorithmus nicht weitergeleitet werden, wird dieses Paket verworfen. Die eingesetzten Routing-Algorithmen sind XY- (XYR), West-First- (WFR), North-Last- (NLR) und Negative-First-Routing (NFR).

XYR ist ein minimales und deterministisches Verfahren, bei dem nacheinander in allen Dimensionen zum Ziel geroutet wird. Für ein Gitternetz bieten sich hierbei X und Y als Dimensionen an. Die positive und negative X-Richtung werden als Ost beziehungsweise West bezeichnet. Analog gelten Nord und Süd für die Y-Richtung. XYR ist ein verteiltes Verfahren, bei dem jeder Router prüft, ob die X-Koordinate des Ziels erreicht ist und gegebenenfalls in Y-Richtung weitergeroutet werden muss, bis das Ziel erreicht ist (siehe Abbildung 2.16).

WFR, NLR und NFR sind verteilte, teilweise adaptive Modifikationen des XYR. Diese weisen implementierungsabhängig eine gegenüber XYR verbesserte Konnektivität auf, wobei die wählbaren Pfade zwischen zwei Knoten nicht zwangsläufig minimal sein müssen. Beim WFR wird ein Paket zuerst so weit wie nötig nach Westen geroutet, falls dies möglich ist. Anschließend wird adaptiv nach Norden, Süden oder Osten geroutet. WFR ist adaptiv, wenn sich der Zielknoten östlich der aktuellen Position befindet. Die Wahl der Richtung ist implementierungsabhängig und für diese Arbeit derart gestaltet,

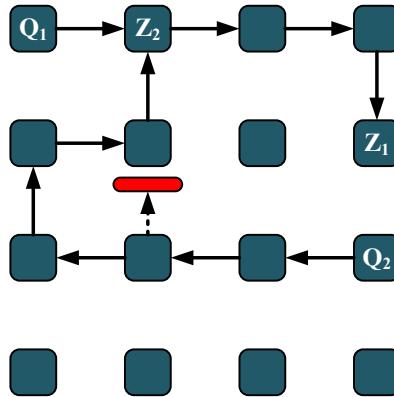


Abbildung 2.16: Beispiel für ausfallfreies XRY (Q_1 nach Z_1) und adaptives WFR im Falle eines defekten Übertragungskanals (Q_2 nach Z_2)

dass ausgefallene Links oder Router umgangen werden. Trotzdem wird ein möglichst kurzer Pfad gewählt (siehe Abbildung 2.16). Dies trifft auch auf NLR und NFR zu. Die Voraussetzung hierfür ist die Kenntnis eines jeden Routers über den Zustand seiner benachbarten Übertragungskanäle und Router. Beim NLR wird zunächst adaptiv nach Osten, Westen und Süden geroutet. Erst im letzten Schritt wird nach Norden geroutet. NLR ist damit nur dann adaptiv, wenn sich das Ziel südlich der aktuellen Position befindet. NFR hingegen sieht ein adaptives Routing zuerst in westlicher und südlicher Richtung und anschließend in nördlicher und östlicher Richtung vor. Damit ist NFR adaptiv, falls das Ziel südwestlich oder nordöstlich der aktuellen Position liegt.

Die vorgestellten Algorithmen ähneln sich prinzipiell stark. Erst die relative Lage von Paketquelle zu -ziel sowie eventuelle Ausfälle auf der resultierenden Route entscheiden darüber, welcher Algorithmus seine Adaptivität nutzen kann und somit zuverlässiger ist. Liegen keinerlei Fehler vor, arbeiten alle Algorithmen aufgrund ihrer Minimalität gleich effektiv. Um die Auswirkung von Defekten auf die Zuverlässigkeit NoC-basierter Systeme zu ermitteln, werden Ausfälle von Routern und Übertragungskanälen im gitternetz- und torusbasierten NoC unter Verwendung der vorgestellten Routing-Algorithmen simuliert. Da die Lage des Fehlers von Bedeutung ist, werden zu diesem Zweck alle möglichen Einzelfehler separat simuliert. Anschließend wird der Durchschnitt gebildet. Ähnlich wird auch für Mehrfachausfälle verfahren, nur das hier aufgrund des Simulationsumfangs lediglich zehn Fehlerkonstellationen ausgewertet werden. Auf die Untersuchung vollständig adaptiver Routing-Algorithmen wird an dieser Stelle verzichtet. Erwartungsgemäß bieten diese aufgrund einer erhöhten Flexibilität bei der Pfadwahl eine zuverlässigere Anbindung einzelner Knoten sowie eine bessere Konnektivität. Allerdings sind solche Algorithmen dafür auch wesentlich komplexer. Eingehende Untersuchungen bezüglich der Auswirkung vollständig adaptiver Algorithmen auf Zuverlässigkeit, Leistungsaufnahme und Ressourcenbedarf sind [C.11] zu entnehmen.

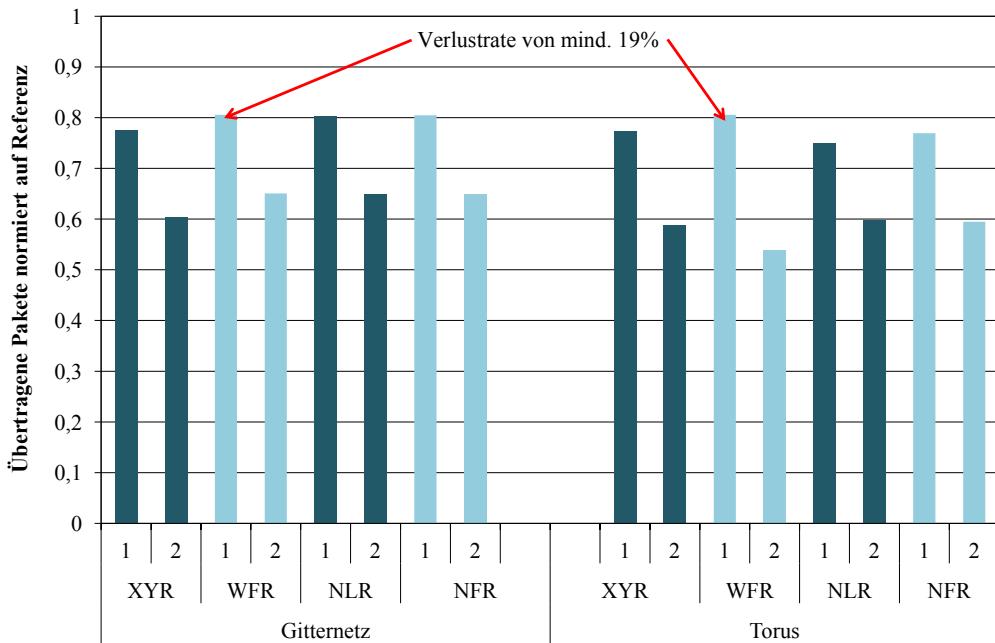


Abbildung 2.17: Auswirkung eines und zweier defekter Router auf die Zuverlässigkeit und Leistungsfähigkeit verschiedener NoC-Topologien und Routing-Algorithmen (normiert auf fehlerfreien Betrieb mit XYR)

Ausfall von Routern In Abbildung 2.17 ist die Auswirkung des Ausfalls eines und zweier Router auf die Anzahl der erfolgreich übertragenen Pakete illustriert. Untersucht wurden eine Gitternetz- und eine Torus-Topologie unter Verwendung der eingangs vorgestellten Routing-Algorithmen. Bereits ein nicht verfügbarer Router sorgt für eine Ausfallrate von mindestens 19 %. Ein Großteil der verworfenen Pakete ist den vom Netzwerk getrennten IPCs zuzuschreiben, welche weder Pakete senden noch empfangen können. Erwartungsgemäß schneiden die teilweise adaptiven Algorithmen besser als reguläres XYR ab. Dieser Vorteil hält sich allerdings in Grenzen und kehrt sich für den Torus im Zusammenhang mit NLR und NFR sogar in einen Nachteil um. Dies ist mit der an den Torus angepassten Implementierung der Routing-Algorithmen in Verbindung mit deren Adaptivität zu erklären. Einerseits werden die zu einer höheren Konnektivität beitragenden globalen Übertragungskanäle nur dann genutzt, wenn die verbleibende Entfernung in X- oder Y-Richtung über den globalen Übertragungskanal kürzer als über den regulären Pfad ist. Andererseits trägt die erhöhte Anzahl der Freiheitsgrade unter Beibehaltung der Minimalität dazu bei, dass eine erhöhte Paketanzahl unzustellbar ist. Offenbar führt die Restriktion, Ausfälle im Netz zu umgehen und dennoch einen möglichst minimalen Pfad zu wählen, in Kombination mit der erhöhten Konnektivität zu diesem Effekt.

Ein Ausfall zweier Router des Netzwerks bewirkt erneut eine deutliche Reduzierung der übertragenen Pakete für beide Architekturen. Abgesehen von WFR in Verbin-

dung mit der Torus-Topologie schneiden alle Routing-Algorithmen geringfügig besser ab als XYR. Der Einbruch von WFR kommt vor allem durch die für diesen Algorithmus ungünstige Lage der defekten Router zustande. Auch fallen die schlechteren Ergebnisse der teilweise adaptiven Algorithmen im Torus verglichen mit deren Gegenstücken im Gitternetz auf. Diese sind auf eine Kombination aus Defektposition, relativer Lage von Quelle zu Ziel, Adaptivität und Minimalität der Pfadwahl zurückzuführen.

Ausfall von Übertragungskanälen Ein ähnliches Bild zeigt sich für den Ausfall eines oder mehrerer Übertragungskanäle. Für das gitternetzbasierte System ergibt sich eine minimale Verlustrate von 9 %, während für den Torus lediglich 5 % zu verzeichnen sind. Die Torus-Topologie weist aufgrund der zusätzlichen globalen Übertragungskanäle in den meisten Fällen eine niedrigere Verlustrate als ihr gitternetzbasiertes Pendant auf. Generell verhalten sich beide Architekturen mit zunehmender Anzahl defekter Übertragungskanäle wie erwartet. Dabei können die teilweise adaptiven Algorithmen innerhalb der jeweiligen Topologie und auch topologieübergreifend Defekte teils deutlich besser kompensieren. Die insgesamt niedrigeren Ausfallraten sowie das bessere Abschneiden des torusbasierten Systems und der teilweise adaptiven Routing-Algorithmen verdeutlichen das Potential zuverlässigkeitsteigernder Maßnahmen. Sie weisen allerdings auch darauf hin, dass durch solche Anpassungen lediglich vereinzelte Ausfälle relativ zuverlässig ausgeglichen werden können. Beispielsweise zieht ein defekter Router bereits einen unerreichbaren IPC und vier nicht verfügbare Übertragungskanäle nach sich. Der Gesamtausfall eines Routers ist zwar nicht das typischerweise zu erwartende Szenario, jedoch verursacht auch ein einzelner defekter Router-Port bereits erhebliche Einschränkungen. So hat ein nicht verfügbarer Router-Port aus der Perspektive des nachfolgenden Routers denselben Effekt wie ein Totalausfall des betroffenen Routers. Räumlich gehäufte Ausfälle sind somit nicht zufriedenstellend kompensierbar.

Die in diesem Abschnitt durchgeführten Untersuchungen zeigen, dass der Ausfall kritischer Komponenten beziehungsweise ein geballtes Auftreten mehrerer Defekte selbst mit verbesserter Vernetzung oder auch teilweise adaptiver Wegfindung nur bedingt kompensiert werden kann. Die Ergebnisse deuten darauf hin, dass räumlich gehäufte Ausfälle, wie sie allgemein durch Verschleiß und Alterung auftreten, starke Einbußen bezüglich der zuverlässigen Kommunikation der On-Chip-Komponenten bedeuten. Beispielsweise war keine der betrachteten Konfigurationen in der Lage, den Ausfall eines oder gar zweier von 16 Routern zufriedenstellend auszugleichen. In beiden Fällen sank die Anzahl der übertragenen Pakete um mindestens 19 %, im schlimmsten Fall sogar um mehr als 40 %. Damit konnte gezeigt werden, dass NoC-basierte Systeme bei dauerhaftem Ausfall von Übertragungskanälen oder Routern nicht in der Lage sind, diese Defekte zu kom-

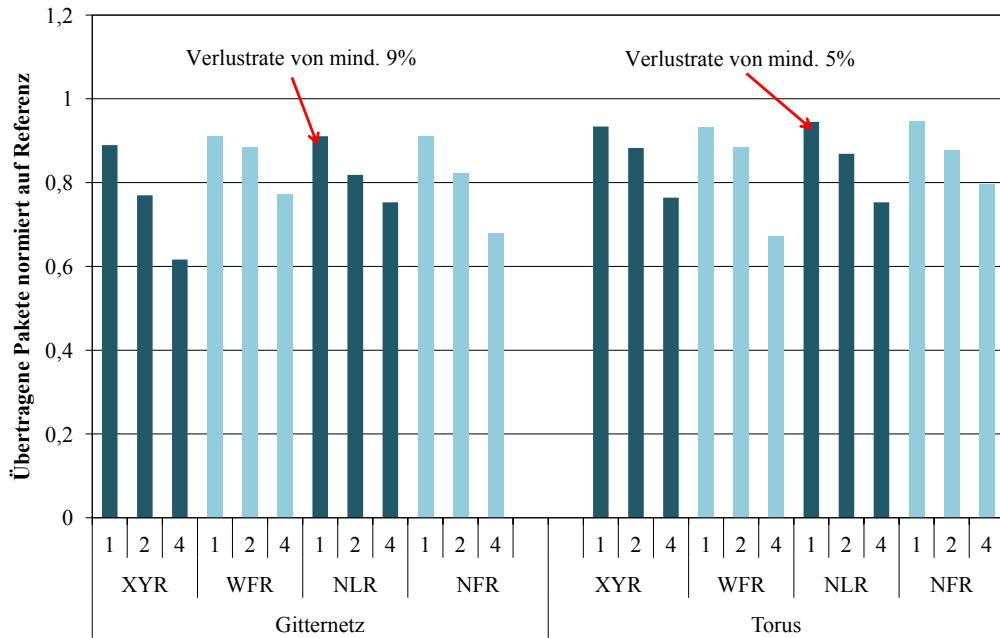


Abbildung 2.18: Auswirkung von ein, zwei und vier defekten Übertragungskanäle auf die Zuverlässigkeit und Leistungsfähigkeit verschiedener NoC-Topologien und Routing-Algorithmen (normiert auf fehlerfreien Betrieb mit XYR)

pensieren. Eine reibungslose Kommunikation zwischen möglichst allen IPCs kann demnach nicht gewährleistet werden. Dies gilt nur für die betrachteten Algorithmen, welche vergleichsweise simpel und lediglich teilweise adaptiv sind. Volladaptive oder quellenbasierte Routing-Algorithmen erzielen voraussichtlich weitaus bessere Ergebnisse, sind allerdings auch dementsprechend aufwändiger und kostenintensiver.

Die Konsequenz ist daher die Erkenntnis, dass das Auftreten von Ausfällen und Fehlern solange wie möglich hinausgezögert und diesem proaktiv vorgebeugt werden muss. Am erfolgversprechendsten ist dabei eine Vorbeugung der langfristigen dynamischen Fehler (siehe Abbildung 2.4), da auf die zugrunde liegenden Effekte durch die Überwachung und Steuerung des Systems positiv Einfluss genommen werden kann. Da die Mehrheit solcher Effekte (z. B. thermischer Stress, EM oder TDDB) durch hohe Temperaturen beschleunigt wird (siehe Abschnitt 2.2.3), ist die Realisierung eines proaktiven temperaturorientierten Managements NoC-basierter Systeme das Ziel dieser Arbeit.

2.4 Ermittlung der Temperatur hochintegrierter Schaltungen

Die Betrachtungen in Abschnitt 2.3.4 haben gezeigt, dass eine Steuerung hochintegrierter Systeme nötig ist, um verschleiß- und alterungsbedingten Defekten vorzubeugen. Andernfalls auftretende Ausfälle und fehlerhaftes Schaltungsverhalten sind nur schwierig

oder gar nicht kompensierbar. Eine besondere Rolle spielt dabei die Temperatur, da diese viele Defektmechanismen beeinflusst (siehe Abschnitt 2.2.3). Um unzulässige Temperaturänderungen frühzeitig zu erkennen und diesen entgegenwirken zu können, ist zunächst eine effiziente und zugleich präzise Methode der Temperaturmessung hochintegrierter Schaltungen notwendig. Die dafür nötigen Temperatursensoren lassen sich auf verschiedene Art und Weise realisieren. Die Ansätze zur Temperaturmessung lassen sich grundlegend in drei Kategorien einteilen [Bla04].

Optische Methoden messen die natürlich emittierte, die reflektierte oder die durch äußere Einflüsse hervorgerufene Strahlung einer elektrischen Schaltung um Rückschlüsse auf deren Temperatur zu ermöglichen.

Realisierungen, die auf physischen Kontakt der Messvorrichtung zur Schaltung zurückgreifen, beruhen auf dem Transfer von thermischer Energie beziehungsweise Wärme von der Schaltung zur Messvorrichtung.

Elektrische Methoden basieren auf der Temperaturabhängigkeit elektrischer Parameter wie Schwell- oder Sperrspannung, Leitungswiderstand oder Schaltgeschwindigkeit. Anhand der Drift solcher Parameter können Rückschlüsse bezüglich des Temperaturverlaufs gezogen werden.

2.4.1 Messung mithilfe von Ringoszillatoren

Der im Folgenden vorgestellte Ansatz zur Messung der Temperatur ist in die Kategorie der elektrischen Methoden einzuordnen und dient im Rahmen dieser Arbeit der Messung des Temperaturverlaufs einer Schaltung. Das Ziel dabei ist der Nachweis der Trägheit der Wärmeausbreitung nach dem Auftreten von Umschaltvorgängen in einem Teil einer digitalen Schaltung. Dieser Nachweis soll exemplarisch anhand eines realen, auf einem Field Programmable Gate Array (FPGA) basierenden Chips erfolgen. Das Vorhandensein einer solchen Latenz ist eine essentielle Grundlage für die laufzeitfähigen Konzepte eines proaktiven Temperaturmanagements sowie eines proaktiven Task-Mappings (siehe Abschnitte 4.2 und 4.3). Diese Konzepte basieren auf der Vorausberechnung beziehungsweise Vorhersage der Temperaturentwicklung aufgrund von Schaltaktivitäten. Dabei wird davon ausgegangen, dass zwischen dem Auftreten einer Schaltaktivität sowie einer entsprechenden Leistungsaufnahme und dem Anstieg der Temperatur eine gewisse Zeit vergeht. Diese Zeitspanne ist idealerweise so bemessen, dass möglichst noch vor der Wärmeausbreitung entsprechende Gegenmaßnahmen eingeleitet werden können und somit die thermische Belastung reduziert werden kann. Dass diese Zeitspanne tatsächlich lang genug ist, soll nachfolgend anhand eines FPGA-basierten Versuchsaufbaus bewiesen werden.

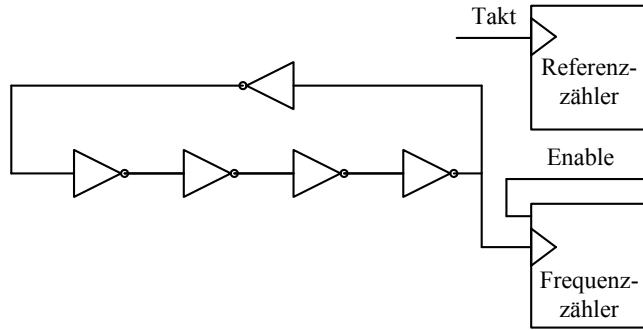


Abbildung 2.19: Schematischer Aufbau eines Ringoszillators sowie eines Frequenzzählers zur Detektion der Änderung von Laufzeitverzögerungen [GWWT12]

Umsetzung

Eine für digitale Schaltungen oft eingesetzte Methode der Temperaturmessung ist die Auswertung der Spannung am pn-Übergang einer Diode [Bla04], welche in die Schaltung integriert ist. Da solche Dioden örtlich gebunden und zahlenmäßig stark beschränkt sind, lässt sich der Temperaturverlauf in von der Diode weiter entfernten Chip-Regionen nur schwer abschätzen. Um dennoch entsprechende Aussagen treffen zu können, eignen sich beispielsweise Ringoszillatoren (RO) als Thermosensoren. Dieser Ansatz der Temperaturmessung erfolgt ausschließlich mithilfe der vorhandenen Chip-Ressourcen und ohne Eingriffe von außen und wurde bereits in [JGB01, VHL⁺05, BLB97, CSZ⁺07] ausgiebig untersucht und validiert. Eine derartige Temperaturüberwachung eines realen Chips kann daher als verlässlich betrachtet werden. Die ROs werden als Logikfunktionen in den konfigurierbaren Logikblöcken (Configurable Logic Blocks (CLB)) des FPGA implementiert und an verschiedenen Stellen auf dem FPGA platziert. Allgemein ist ein RO eine Schaltung, welche durch ihre interne Logikfunktion mit einer gewissen Frequenz schwingt. Eine solche Schaltung kann mithilfe einer ungeraden Anzahl von Invertern sowie zweier Zähler realisiert werden. FPGA-spezifisch werden dazu die in den CLBs vorhandenen Lookup Tables (LUTs) genutzt, welche mithilfe von Schaltmatrizen verbunden sind. Der schematische Aufbau eines RO ist in Abbildung 2.19 illustriert [GWWT12].

Der Referenzzähler generiert eine fest definierte Zeitspanne, welche auf einer bekannten Taktrate basiert. Diese Referenzdauer wird genutzt, um den zweiten Zähler zu steuern, welcher die Anzahl der Durchläufe des Signals durch den Ring während der Referenzdauer festhält. Die Laufzeitverzögerung des Signals entsteht dabei zwangsläufig durch das Weiterleiten des Signals von LUT zu LUT in den Schaltmatrizen. Ändert sich mit der Zeit nun die Anzahl der registrierten Durchläufe, bedeutet dies eine Veränderung der Verzögerung. Die auf diese Weise entstehende Frequenzdrift des RO ist hauptsächlich auf die Temperaturabhängigkeit der Schwellspannung V_{Th} und der Ladungsträgermobilität μ (siehe Abschnitt 2.2.3) sowie die anliegende Versorgungsspan-

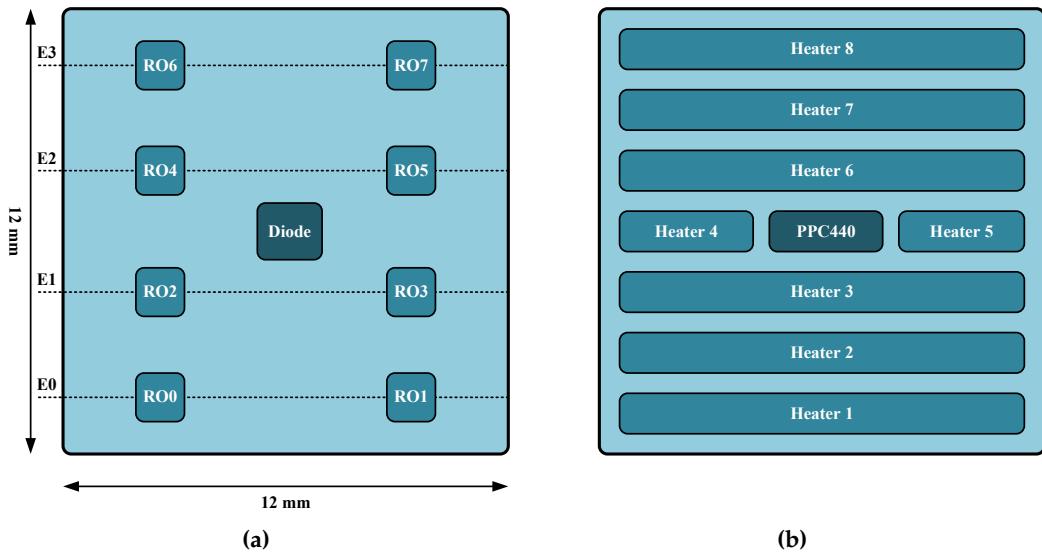


Abbildung 2.20: Virtex5-FPGA: (a) Lage der Temperatursensoren sowie der Referenz-Diode, (b) Platzierung der Test-Heater

nung zurückzuführen. Damit lassen sich relativ genau Aussagen über die Temperatur treffen. Darüber hinaus können mit entsprechend angepassten ROs Rückschlüsse auf die Signalintegrität sowie die Alterung und den Verschleiß einer Schaltung gezogen werden, da Crosstalk-Effekte und alterungsbedingte Verschlechterungen des Schaltungs-Timings ebenfalls Einfluss auf die Frequenzdrift haben. Dabei muss allerdings berücksichtigt werden, dass auch die ROs Alterung und Verschleiß unterworfen sind.

Durchführung der Messungen

Für die folgenden Untersuchungen wird ein in 65 nm gefertigtes Virtex5-FPGA genutzt, welches über eine interne Diode sowie einen integrierten Mikroprozessor verfügt (siehe Abbildungen 2.20(a) und 2.20(b)). Die im Zentrum des Chips gelegene Diode dient der Temperaturmessung und fungiert als Referenzwert für die mithilfe der ROs durchgeführten Messungen. Bei dem Prozessor handelt es sich um einen PowerPC 440, welcher die Auswertung der von den ROs generierten Daten übernimmt. Die Größe des eigentlichen FPGA-Chips wird mit 12×12 mm angenommen [ABG12], die Ausgangstemperatur liegt laut Diode bei 46°C . Insgesamt werden acht Instanzen des RO, im weiteren Verlauf als Thermosensoren bezeichnet, so auf dem Chip platziert (siehe Abbildung 2.20(a)), dass diese sich auf vier horizontale Ebenen verteilen. Um gezielt Temperaturanstiege in einzelnen Bereichen des Chips hervorzurufen, wurden mehrere als Heater bezeichnete Logikblöcke definiert (siehe Abbildung 2.20(b)). Diese bestehen aus Latches, welche einzeln ansteuerbar sind und durch Umschaltvorgänge Temperaturanstiege verursachen.

Tabelle 2.3: Genauigkeit der ringoszillatortbasierten Temperatursensoren bezogen auf die an der Diode gemessene Temperatur

Abweichung	RO0	RO1	RO2	RO3	RO4	RO5	RO6	RO7
$\leq 2,0 \text{ }^{\circ}\text{C}$	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
$\leq 1,0 \text{ }^{\circ}\text{C}$	95 %	95 %	97 %	93 %	98 %	97 %	97 %	97 %
$\leq 0,5 \text{ }^{\circ}\text{C}$	85 %	86 %	88 %	86 %	88 %	85 %	92 %	86 %
$\leq 0,1 \text{ }^{\circ}\text{C}$	34 %	32 %	29 %	36 %	37 %	37 %	32 %	29 %

Kalibrierung Zunächst ist eine individuelle Kalibrierung der Thermosensoren wegen positionsabhängiger Schwankungen der Schaltungsparameter sowie der Versorgungsspannung nötig. Andernfalls entstehen falsche Korrelationen zwischen gemessener Signalfrequenz und zugeordneter Temperatur für die einzelnen Thermosensoren. Zu diesem Zweck wird der Frequenzverlauf der Thermosensoren während der Aufheizung des gesamten Chips durch die Aktivierung sämtlicher Heater aufgezeichnet und mit den Temperaturwerten der Diode abgeglichen. Dieser Abgleich ist trotz der räumlichen Entfernung zwischen der Diode und den Thermosensoren zulässig, da eine gleichmäßige Aufheizung vorausgesetzt wird. Während der gesamten Kalibrierung wurden für jeden Thermosensor 60 Messungen vorgenommen. Die Genauigkeit der einzelnen Sensoren bezogen auf die Diode ist in Tabelle 2.3 dargestellt. Nach der Kalibrierung liegt demnach die Ungenauigkeit aller Sensoren unter $\pm 2 \text{ }^{\circ}\text{C}$. Der Großteil aller Messwerte weist sogar eine Genauigkeit von $\pm 0.5 \text{ }^{\circ}\text{C}$ auf. Lediglich ein Wert von $\pm 0.1 \text{ }^{\circ}\text{C}$ kann nicht zufriedenstellend erreicht werden. Angesichts einer absoluten Genauigkeit der Diode von $\pm 4 \text{ }^{\circ}\text{C}$ [GWWT12] ist dieses Resultat akzeptabel.

Messung und Auswertung Für die Überprüfung der Annahme, dass die auf eine Schaltaktivität folgende Wärmeausbreitung eine signifikante Zeitspanne benötigt, wird nun gezielt die obere Region des FPGA-Chips mithilfe der Heater 7 und 8 aufgeheizt. Dieser Vorgang ist in Abbildung 2.21 für eine Dauer von 50 s dargestellt. Durch die auf einen Teilbereich des FPGA beschränkte Aufheizung macht sich zunächst ein Temperaturanstieg im Bereich der ROs RO4 bis RO7 bemerkbar (siehe Abbildung 2.21(b)). Im weiteren Verlauf breitet sich die Wärme in die restlichen Bereiche des FPGA aus, sodass auch die ROs RO0 bis RO3 einen Temperaturanstieg registrieren (siehe Abbildung 2.21(c)). Da die Heater weiterhin Wärme erzeugen, erhitzt sich besonders der obere Bereich des FPGA bis auf etwa $90 \text{ }^{\circ}\text{C}$, wobei diese Wärme sich mit einer gewissen Verzögerung in die unteren Regionen des FPGA ausbreitet (siehe Abbildungen 2.21(d) bis 2.21(f)).

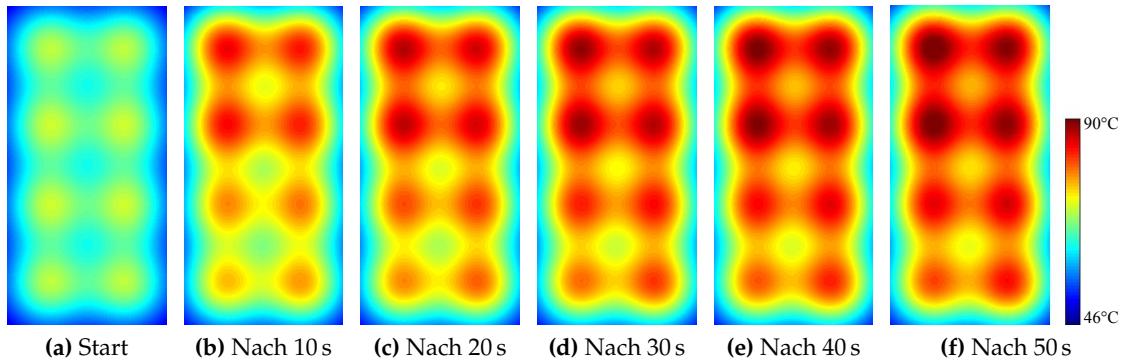


Abbildung 2.21: Temperaturausbreitung auf dem FPGA über eine Dauer von 50 s bei Aktivierung der Heater 7 und 8 (Ausgangstemperatur: 46 °C)

Wie vermutet erhöht sich die Latenz der Temperaturausbreitung mit steigender Entfernung der Sensoren von der Wärmequelle. Dabei heben sich die einzelnen Ebenen mit steigender Temperaturerhöhung immer deutlicher voneinander ab. Die in Abbildung 2.22 dargestellten Resultate beziehen sich auf die horizontalen Ebenen E0 bis E3, welche durch jeweils zwei Sensoren gebildet werden. Werden am Ort der Leistungsaufnahme (d. h. Ebene E3) für eine Temperaturerhöhung von 0.5 °C noch 55 ms benötigt, sind es für 2.0 °C bereits 355 ms. Eine Erhöhung im zweistelligen Bereich nimmt bereits mehrere Sekunden in Anspruch. Dieses Gesamtbild lässt auf einen exponentiellen Verlauf für jede einzelne Ebene schließen.

Auch die räumliche Ausbreitung weist mit größer werdender Entfernung vom Ort der Wärmeabstrahlung eine deutlich zunehmende Verzögerung auf. Dies äußert sich bereits bei vergleichsweise geringen Temperaturänderungen von 0.5 °C und 1.0 °C. So benötigt eine Erhöhung der Temperatur um 1.0 °C in Entfernung der Ebene E2 bereits 66 ms länger als am Ursprung der Temperaturerhöhung. Für die Ebenen E1 und E0 wird für diesen Wert bereits doppelt so viel Zeit benötigt. Mit steigenden Temperaturerhöhungen deutet sich auch hier ein exponentieller Trend an, sodass eine Temperaturerhöhung um signifikante Werte für weiter entfernte Chip-Gebiete (Ebenen E1 und E0) zweieinhalb- bis dreimal solange dauert. Für eine Erhöhung um 20.0 °C werden aber auch weitaus höhere Werte erreicht. Eine solche Temperaturerhöhung nimmt selbst in näherer Umgebung der Wärmequelle (d. h. Ebene E2) mehrere Sekunden in Anspruch. Zwar sind keine exakten Angaben für die Abstände zwischen den einzelnen Ebenen bekannt, aber aufgrund der Kantenlänge des Chips von 12 mm sollten diese sich im Bereich von wenigen Millimetern bewegen. Dass solche Dimensionen auch praktische Relevanz haben, wird am Beispiel aktueller Intel-Chips [Int13] deutlich.

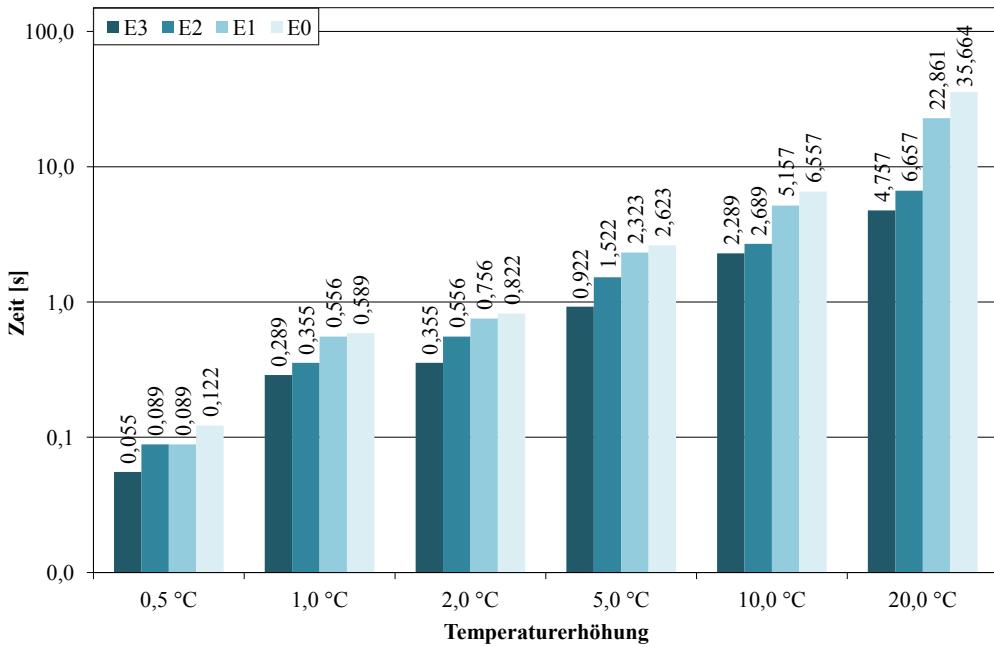


Abbildung 2.22: Dauer der Temperaturausbreitung für verschiedene Zonen des FPGA bei Aktivierung der Heater 7 und 8 (logarithmische Darstellung)

In diesem Abschnitt konnten zwei wichtige Aspekte bestätigt und verdeutlicht werden. Erstens benötigt die durch die Leistungsaufnahme einer Schaltung verursachte Wärmeausbreitung am Ort der Aufnahme für messbare Werte mindestens mehrere hundert Millisekunden. Dieser Trend nimmt mit wachsendem Temperaturanstieg exponentiell zu. Zweitens ist auch die Ausbreitung der abgestrahlten Wärme in umgebende und weiter entfernte Chip-Regionen einer deutlichen Verzögerung unterworfen. Somit vergehen teils mehrere Sekunden bis ähnliche Temperaturwerte erreicht werden, wie sie am Ursprung der Abstrahlung vorherrschen. Mit dem Nachweis solcher Ausbreitungsverzögerungen ist die Grundlage für eine Untersuchung der Machbarkeit eines proaktiven temperaturorientierten Systemmanagements und Task-Mappings gesichert.

2.5 Folgerung und Problemformulierung

Die in diesem Kapitel aufgeführten technologischen Trends sowie die damit zusammenhängenden Problemstellungen und Sachverhalte bilden den Ausgangspunkt für den weiteren Verlauf dieser Arbeit.

Zunächst machen die aufgezeigten technologischen Entwicklungen und die sich ergebenden problematischen Aspekte ein Umdenken erforderlich. Rein leistungsorientierte Entwurfsmethoden werden somit zunehmend durch Konzepte verdrängt, welche auch Zuverlässigkeit, Robustheit und Energieeffizienz berücksichtigen oder diese gar

priorisieren. So haben neben Leistungsfähigkeit und Leistungsaufnahme auch Faktoren wie die Temperaturentwicklung stark an Bedeutung gewonnen (siehe Abschnitte 2.1.2 und 2.2.3), da diese maßgeblichen Einfluss auf die Lebensdauer und die Funktionsfähigkeit hochintegrierter Systeme haben. Die enge Verknüpfung dieser Faktoren untereinander erschwert einen entsprechenden Systementwurf zusätzlich. Die Neuausrichtung der Entwurfsmethodik muss ebenfalls auf den Betrieb hochintegrierter Systeme übertragen werden, um langfristig eine zuverlässige Funktions- und Leistungsfähigkeit zu gewährleisten. An dieser Stelle sind entsprechende Managementkonzepte und Mechanismen erforderlich, da sich einige Problemstellungen bezüglich Leistungsaufnahme, Alterung und Verschleiß nicht ausschließlich durch Maßnahmen während der Entwurfsphase lösen lassen.

Darüber hinaus bedeuten steigende Integrationsdichten zunehmend komplexe Systeme. Dies stellt ein wachsendes Problem für die Entwurfseffizienz und -produktivität dar, da beispielsweise auch der Planungs- und Koordinationsaufwand in hohem Maße zunimmt (siehe Abschnitt 2.1.1). Der Grund hierfür ist die steigende Anzahl sowie die Heterogenität der auf einem einzelnen Chip integrierbaren Systemkomponenten. Besserung verspricht diesbezüglich die Verlagerung des Entwurfs weg von klassischen verarbeitungsorientierten Architekturen mit zentralem Kommunikationsmedium hin zu modularen, kommunikationsorientierten Architekturen mit dezentralem Medium. Damit kann der steigenden Anzahl einzelner zu vernetzender Komponenten sowie deren Anforderungen an die Kommunikationsinfrastruktur Rechnung getragen werden. Networks-on-Chip (NoCs) stellen in diesem Zusammenhang ein vielversprechendes Konzept dar (siehe Abschnitt 2.3.1). Diese zeichnen sich neben ihrer Modularität auch durch eine gute Skalierbarkeit und eine hohe Leistungsfähigkeit aus.

Das Ziel dieser Arbeit ist es nun die Zusammenführung der aufgezeigten Trends voranzutreiben, sodass sich im Grunde drei wichtige Kernthemen formulieren lassen. Erstens steht die Behandlung temperaturbezogener Aspekte im Mittelpunkt, da auf diese Weise alterungs- und verschleißbedingte Defekte hinausgezögert werden können. Zweitens konzentrieren sich die Untersuchungen dabei auf entsprechende Konzepte für ein Temperaturmanagement zur Laufzeit betrachteter Systeme. Drittens stützen sich die entwickelten Lösungsansätze auf NoC-basierte Architekturen.

Im Detail bedeutet dies, dass ausgewählte Konzepte bezüglich ihrer Einsetzbarkeit für ein temperaturorientiertes Laufzeitmanagement NoC-basierter Systeme untersucht werden. Dies soll möglichst unter Ausnutzung der durch das NoC gegebenen Ressourcen und architekturbedingten Vorteile geschehen. Abbildung 2.23 illustriert den generellen Aufbau eines solchen zuverlässigkeitsorientierten Systemmanagements und verdeutlicht dessen Einordnung in ein bestehendes System. Das dynamische Laufzeitmanagement findet auf der Architekturebene statt, da gezielt Systemkomponenten überwacht

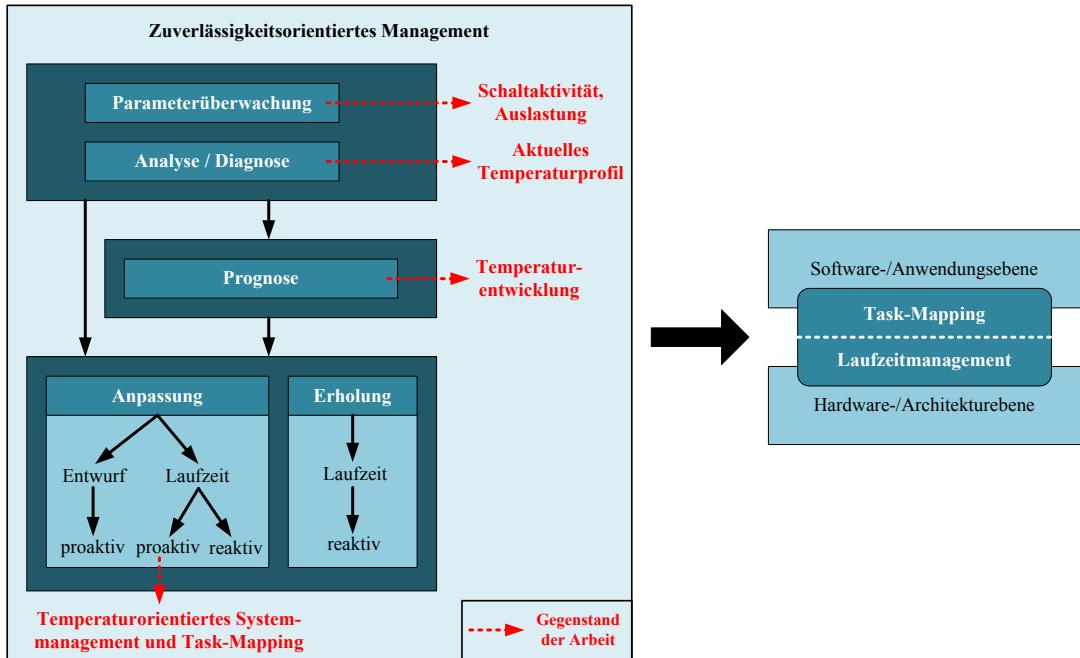


Abbildung 2.23: Allgemeiner Aufbau eines mehrschichtigen, zuverlässigkeitsoorientierten Systemmanagements und Einordnung in ein bestehendes System

und gesteuert werden. Die Verarbeitung und Auswertung der Überwachungsinformationen ist jedoch auch als Software vorstellbar. Das dynamische Task-Mapping findet hingegen auf der Anwendungsebene statt, da hierfür kein Wissen bezüglich hardwarenaher Parameter wie Temperatur oder Schaltaktivität nötig ist. Vielmehr sind Informationen wie Auslastung oder Task-Verteilung erforderlich, welche software-seitig ermittelbar sind. Mithilfe dieser Daten können auszuführende Anwendungen zur Laufzeit des Systems auf den einzelnen Komponenten platziert werden. Auf diese Weise kann der aktuelle Zustand des Systems, dies betrifft unter anderem die vorherrschende Temperaturverteilung, bei der Platzierung einzelner Tasks berücksichtigt werden. Der Bezug zu NoCs ergibt sich aus der Überwachung der Aktivitäten der einzelnen Infrastrukturkomponenten sowie der Lastsituation der durch das NoC verbundenen Systemkomponenten. Des Weiteren wird für die Übertragung der Überwachungsdaten und Managementinstruktionen das NoC genutzt, in welches das gesamte Managementsystem eingebettet ist. Die Steuerungsinstruktionen selbst umfassen neben der dynamischen Anpassung der Leistungsfähigkeit eine Abschaltung einzelner Systemkomponenten sowie eine dynamische Verteilung einzelner Tasks auf diesen.

3 Temperaturmodell

In diesem Kapitel wird das Temperaturmodell vorgestellt, welches als Basis für die in den Abschnitten 4.2 und 4.3 eingeführten proaktiven Konzepte für ein temperaturorientiertes Systemmanagement und Task-Mapping dient. Um den zeitlichen Vorteil solch proaktiver Ansätze gegenüber ihren herkömmlichen reaktiven Gegenstücken nutzen zu können, ist eine schnelle und zugleich ausreichend genaue Modellierung des Temperaturverhaltens der zu überwachenden Systeme notwendig. Nur auf diese Weise kann gewährleistet werden, dass für zu erwartende Temperaturänderungen möglichst vorausschauend die richtigen Maßnahmen getroffen werden.

Aus diesem Grund wird zunächst in Abschnitt 3.1 das entwickelte Modell vorgestellt und bezüglich der zugrunde liegenden Mechanismen sowie der Konfigurations- und Anpassungsmöglichkeiten eingehend erläutert. Anschließend werden in Abschnitt 3.2 die Genauigkeit und die Modellierungsgeschwindigkeit des Modells untersucht und bewertet. Die Bewertung geschieht sowohl qualitativ als auch anhand durchgeföhrter Simulationen. Als Vergleichspartner dient ein identisches, mithilfe von SPICE (Simulation Program with Integrated Circuit Emphasis) realisiertes Modell. Dieses ist als Referenz zwar bereits physikalisch exakt, aber aufgrund der Beschaffenheit SPICE-basierter Modelle nicht für einen Einsatz zur Laufzeit geeignet.

3.1 Realisierung

Die grundlegende Idee des Temperaturmodells ist es, das dynamische, thermische Verhalten hochintegrierter Schaltkreise durch die Nutzung äquivalenter elektrischer RC-Glieder zu modellieren. Dies ist aufgrund eines Dualismus zwischen dem Fluss elektrischer und thermischer Energie möglich [Kru00], welcher nachfolgend erläutert wird.

3.1.1 Dualismus elektrischer und thermischer Energie

Die dem Dualismus unterliegenden Mechanismen folgen den Gesetzmäßigkeiten der Thermodynamik. Demnach kann ein Wärmefluss innerhalb einer Schaltung als ein Stromfluss durch einen elektrischen Widerstand beschrieben werden. Dieser Strom verursacht einen Spannungsabfall über den elektrischen Widerstand, welcher äquivalent zu einem Temperaturunterschied über den thermischen Widerstand ist (siehe Abbildung 3.1). Des

Tabelle 3.1: Dualismus zwischen dem Fluss elektrischer und thermischer Energie: elektrische und äquivalente thermische Parameter

Elektrischer Parameter	Einheit	Thermischer Parameter	Einheit
Strom	I [A]	Leistung, Wärmefluss	\dot{Q} [W]
Spannung	U [V]	Temperaturunterschied	ΔT [K]
elektr. Widerstand	R_{el} [V/A]	therm. Widerstand	R_{th} [K/W]
elektr. Kapazität	C_{el} [As/V]	therm. Kapazität	C_{th} [J/K]
elektr. Zeitkonstante	τ_{el} [s]	therm. Zeitkonstante	τ_{th} [s]

Weiteren repräsentieren elektrische Kondensatoren die Wärmekapazität eines Schaltungsteils, also die durch diese Komponente der Schaltung maximal aufnehmbare Wärmemenge. Elektrische Widerstände hingegen repräsentieren den thermischen Widerstand für den Wärmefluss in eine bestimmte Richtung, beispielsweise von einem Teil der Schaltung in einen anderen. Die zeitliche Verzögerung der Erwärmung wird durch eine thermische Zeitkonstante τ_{th} beschrieben, welche äquivalent zur elektrischen Zeitkonstante ist. Im Detail ist τ_{th} ein Maß für die Dauer der Auf- und Entladung einer thermischen Kapazität und ergibt sich aus dem Produkt des thermischen Widerstandes und der thermischen Kapazität einer Komponente. Als Faustregel für die Aufladung gilt, dass nach Ablauf von $1 \cdot \tau_{th}$ eine thermische Kapazität zu etwa 63 % aufgeladen ist. Erst nach $5 \cdot \tau_{th}$ gilt die Kapazität als vollständig aufgeladen. Äquivalent dazu ist eine solche Kapazität nach $1 \cdot \tau_{th}$ auf 37 % und nach $5 \cdot \tau_{th}$ nahezu vollständig entladen. Die Analogien sind in Tabelle 3.1 zusammengefasst.

In Abbildung 3.1 sind beispielhaft die äquivalenten Widerstände mit den Parametern für eine geometriebasierte Berechnung dargestellt. Der elektrische Widerstand wird nach Gleichung 3.1 aus der Strecke l , welche der Strom durchfließt, sowie der elektrischen Leitfähigkeit σ und der vom Strom durchflossenen Querschnittsfläche A berechnet. Analog dazu ergibt sich der thermische Widerstand nach derselben Gleichung. Lediglich σ wird durch die thermische Leitfähigkeit λ ersetzt. λ beträgt für Silizium $100 \frac{W}{m \cdot K}$ und für Kupfer $400 \frac{W}{m \cdot K}$. Die über den Widerständen entstehenden Spannungsabfälle beziehungsweise Temperaturunterschiede werden nach Gleichung 3.3 berechnet. Elektrische und thermische Kapazitäten werden nach Gleichung 3.2 berechnet, wobei C_{el} den Be-

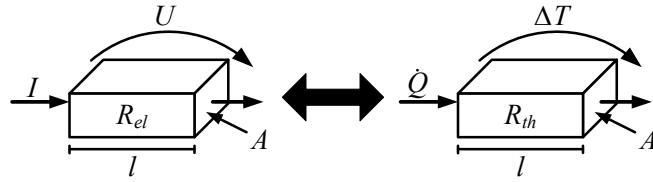


Abbildung 3.1: Äquivalenz zwischen elektrischem und thermischem Widerstand

rechnungsvorschriften für einen Plattenkondensator folgt. Die thermische Kapazität, also das Wärmefassungsvermögen eines Körpers, berechnet sich hingegen als das Produkt aus Volumen und spezifischer Wärmekapazität c des betrachteten Körpers. c beträgt für Silizium $1,75 \cdot 10^6 \frac{\text{J}}{\text{m}^3}$ und für Kupfer $3,55 \cdot 10^6 \frac{\text{J}}{\text{m}^3}$.

$$R_{el} = \frac{l}{\sigma \cdot A} \Leftrightarrow R_{th} = \frac{l}{\lambda \cdot A} \quad (3.1)$$

$$C_{el} = \epsilon \cdot \frac{A}{l} \Leftrightarrow C_{th} = c \cdot l \cdot A \quad (3.2)$$

$$U = R_{el} \cdot I \Leftrightarrow \Delta T = R_{th} \cdot \dot{Q} \quad (3.3)$$

3.1.2 Stand der Technik

Die Modellierung des thermischen Verhaltens komplexer Systeme auf Basis des in Abschnitt 3.1.1 vorgestellten Dualismus wurde bereits mehrfach unter verschiedenen Aspekten untersucht. In [SSH⁺03] werden äquivalente RC-Glieder genutzt, um das thermische Verhalten eines kompletten Chips zu modellieren. Neben dem eigentlichen Chip werden auch ein Kühlkörper und die Abführung der Wärme über diesen berücksichtigt. Durch eine variable Modellauflösung werden Kompromisse zwischen Modellierungsgenauigkeit und Berechnungsgeschwindigkeit ermöglicht. Die Temperatur der einzelnen funktionalen Module wird auf Basis von zuvor ermittelten Durchschnittswerten für deren Leistungsaufnahme berechnet.

In [SPKJ04] wird dieser Ansatz modifiziert, um das thermische Profil NoC-basierter Systeme zu modellieren. Des Weiteren wird das zugrunde liegende Modell um Wärmeabstrahlungswinkel ergänzt. Die eigentliche Simulation des Temperaturprofils gliedert sich in drei Phasen. Zunächst wird mithilfe einer funktionalen Simulation die Netzwerkaktivität der gesamten betrachteten Dauer aufgezeichnet. Auf Grundlage dieser Statistiken werden anschließend die Leistungsaufnahme abgeschätzt und das Temperaturprofil berechnet.

In [LCN⁺09, TCST10] werden dagegen auf äquivalenten RC-Gliedern basierende SPICE-Netzlisten erzeugt, um thermische Eigenschaften von On-Chip-Netzwerken mit

verschiedenen Genauigkeitsstufen zu modellieren. Dies geschieht auf Basis von Aktivitätsprofilen, welche zuvor in funktionalen Simulationen ermittelt wurden und dem SPICE-Modell für die Berechnung der Temperaturverteilung übergeben werden. Das SPICE-Modell zeichnet sich durch eine hohe Genauigkeit aus, hat aber den Nachteil einer unter Umständen hohen Berechnungszeit und einer starren, sequentiellen Abfolge von Aktivitätsermittlung und Temperaturberechnung.

Dies ist auch der Hauptkritikpunkt an den übrigen vorgestellten Referenzen. Deren Modelle sind derart gestaltet, dass lediglich eine sequentielle Abarbeitung von Aktivitätsaufzeichnung und Temperaturberechnung für eine im Vorfeld festgelegte Dauer möglich ist. Dies bedeutet, dass ein Einsatz dieser Konzepte für ein laufzeitfähiges temperaturorientiertes Systemmanagement unmöglich ist, da immer erst die Aktivitätsprofile der Gesamtdauer vorliegen müssen. Erst dann kann der Temperaturverlauf ermittelt werden. Weiterhin greifen diese Konzepte auf externe Werkzeuge wie Wattch [BTM00] oder Orion [WZPM02] zurück, um die für eine Temperaturmodellierung benötigten Statistiken bezüglich der Leistungsaufnahme zu ermitteln.

Das im Folgenden vorgestellte Modell behebt diese Missstände und ermöglicht einen parallelen Ablauf von funktionaler Simulation sowie dazugehöriger Aktivitätsüberwachung und der darauf basierenden Temperaturberechnung. Auf externe Werkzeuge wird dabei vollständig verzichtet. Die für die Berechnung der Temperatur benötigten Abschätzungen bezüglich der Leistungsaufnahme werden den Untersuchungen vorhandener Arbeiten entnommen.

3.1.3 Implementierung und Simulationsablauf

Um eine parallele Simulation des funktionalen und des thermischen Verhaltens zu ermöglichen, wird die Gesamt simulationsdauer in gleich lange Zeitabschnitte eingeteilt. Diese Abschnitte werden fortan als Simulationsperiode T_S bezeichnet. Nach Ablauf einer Simulationsperiode werden die Aktivitätsstatistiken ausgewertet. Zusätzlich werden die Ergebnisse der Temperaturberechnungen in Form eines Temperaturprofils ausgegeben. Auf diese Weise kann bereits während der funktionalen Simulation eine parallele Berechnung des Temperaturverlaufs erfolgen. Wird T_S kurz genug gewählt, kann zudem eine bis zu einem gewissen Grad vorausschauende Berechnung des Temperaturverlaufs erfolgen. Möglich ist dies durch die in Abschnitt 2.4.1 untersuchte Trägheit der Wärmeausbreitung nach dem Auftreten einer Schaltaktivität. Mithilfe dieser Methode lassen sich bei ausreichender Genauigkeit und Leistungsfähigkeit des Modells effektive Lösungen für ein laufzeitfähiges, temperaturorientiertes Systemmanagement konzipieren.

Implementierung

Um ein Modell mit den beschriebenen Vorteilen gegenüber herkömmlichen Methoden zu ermöglichen, wurde eine bereits bestehende Simulationsumgebung für die funktionale Simulation NoC-basierter Systeme entsprechend erweitert. Die Simulationsumgebung basiert auf der C++-Spracherweiterung SystemC [Sys13] und dem Transaction-Level-Modeling-Standard (TLM) in der Version 2.0 sowie der dazugehörigen SystemC-Bibliothek [TLM13]. Die Umgebung zeichnet sich durch eine stark erhöhte Leistungsfähigkeit gegenüber auf Register Transfer Level (RTL) durchgeführten Simulationen bei identischer Funktionalität aus. Des Weiteren bietet sie flexible Konfigurationsmöglichkeiten bezüglich verschiedener Parameter wie beispielsweise der NoC-Dimensionen, der Flitbreite oder der Simulationsdauer. Die Injektionsrate der in das NoC eingebundenen IPCs, also die Rate der Paketgenerierung, lässt sich ebenso festlegen wie die minimale und maximale Länge der generierten Pakete. Die Topologie ist auf gitternetzbasierte NoCs beschränkt.

Die Leistungsfähigkeit des Simulators liegt in der großen Anzahl von Freiheitsgraden für den SystemC-gestützten Systementwurf begründet. Diese Freiheitsgrade ermöglichen abstrakte Systemmodelle mit beeinflussbarer Beschreibungsgenauigkeit bezüglich der Systemstruktur, des Zeitmodells, der Funktionalität, der Datenorganisation und des Kommunikationsprotokolls. Mithilfe des TLM-Standards lassen sich insbesondere Datentransaktionen innerhalb des modellierten Systems abstrahieren, anstatt diese aufwändig auf RTL-Ebene mithilfe von Signalen zu modellieren. Einbußen bezüglich der zeitlichen Genauigkeit müssen dabei nicht hingenommen werden. Der eingesetzte Simulator bietet somit eine taktgenaue Simulation der Funktionalität der einzelnen NoC-Komponenten sowie der Kommunikation zwischen diesen. Datentransaktionen zwischen den unterschiedlichen Komponenten werden dabei durch Funktionsaufrufe abstrahiert, während wait-Anweisungen das Voranschreiten der Zeit realisieren. Darüber hinaus weisen die einzelnen Simulationsprozesse eine dynamische Sensitivität gegenüber Ereignissen und Transaktionen auf, anstatt einer statischen Sensitivität gegenüber festgelegten Signalen (z. B. Taktsignal) zu folgen. Der zeitliche Simulationsaufwand kann damit deutlich reduziert werden, da Prozesse nur dann ausgeführt werden, wenn dies aus funktionaler Sicht nötig ist [GWG⁺13].

Das Temperaturmodell ist in die vorgestellte Simulationsumgebung integriert und basiert auf der SystemC-eigenen Analog-Mixed-Signal (AMS)-Bibliothek [AMS13]. Diese Bibliothek erlaubt es, das Modell nahtlos in die bestehende Umgebung einzufügen. Darüber hinaus wird die software-basierte Modellierung elektrischer Parameter und Abläufe in der Form ermöglicht, in der dies für die Beschreibung des eingangs vorgestellten Dualismus elektrischer und thermischer Parameter nötig ist (siehe Tabelle 3.1). Im Einzelnen wird auf die zu den Electrical Linear Networks (ELN) gehörenden Bibliotheksbestandtei-

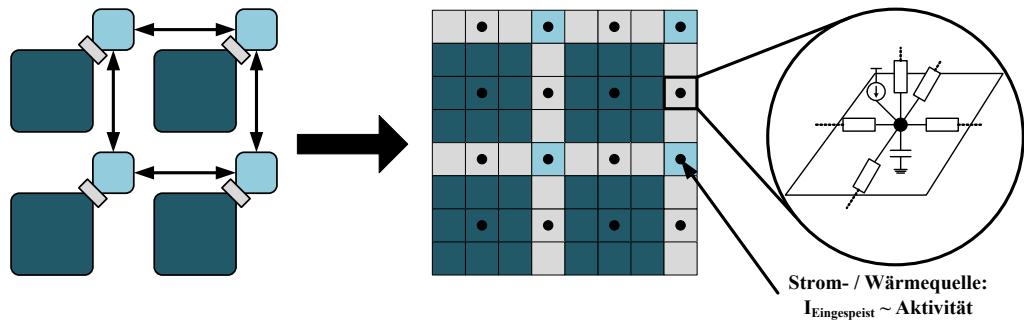


Abbildung 3.2: Abbildung eines 2×2 NoC auf ein äquivalentes RC-Netz: jeder Komponente wird eine Stromquelle zugewiesen, über welche ein zur Schaltaktivität äquivalenter Strom eingespeist wird

le zurückgegriffen. Diese erlauben neben der Beschreibung des elektrischen Verhaltens primitiver Basiselemente (z. B. Widerstände oder Kondensatoren) auch die zeitkontinuierliche Modellierung des Verhaltens der resultierenden linearen elektrischen Netzwerke.

Für die Erstellung des Temperaturmodells mithilfe von SystemC AMS wird das NoC-basierte System zunächst in Kacheln eingeteilt. Jede Kachel besteht aus mehreren elektrischen Widerständen, einem Kondensator und gegebenenfalls einer Stromquelle. Diese Abbildung des Systems auf ein in Kacheln eingeteiltes Netz aus Widerständen, Kapazitäten und Stromquellen ist in Abbildung 3.2 dargestellt. Die Widerstände modellieren den Wärmewiderstand in die jeweilige Richtung, den ein etwaiger Wärmefluss zu überwinden hat. Der Kondensator repräsentiert die Wärmemenge, die der betreffende Teil des Chips aufnehmen kann bevor die Wärme in benachbarte Kacheln abgeleitet wird. Jede Systemkomponente erhält zusätzlich eine Stromquelle, die in der Kachel platziert wird, welche möglichst zentral in der jeweiligen Komponente liegt. Der in diese Stromquelle eingespeiste Strom ist proportional zu der für die betreffende Komponente beobachteten Schaltaktivität und repräsentiert die durch die Aktivität erzeugte Wärme (siehe Tabelle 3.1). Der Strom berechnet sich nach Gleichung 3.4.

$$I = \frac{(\sum Trans_{0 \rightarrow 1} \cdot E_{Trans})}{\Delta t} + P_{Leck} \quad (3.4)$$

$Trans_{0 \rightarrow 1}$ ist die Anzahl der Bitübergänge von 0 auf 1 für die beobachtete Komponente (d. h. IPC, Router oder Übertragungskanal). E_{Trans} ist die für einen Übergang benötigte Energie und Δt ist die Zeitspanne, während der die Übergänge beobachtet wurden. Es werden vereinfachend lediglich $0 \rightarrow 1$ -Übergänge betrachtet, da nur diese einen Ladestrom und somit eine Energieaufnahme verursachen (siehe Abschnitt 2.1.2).

Für Übertragungskanäle wird E_{Trans} mit $11,62 \text{ fJ}$ für eine 65 nm -Technologie bei einer Leitungslänge von $200 \mu\text{m}$ und zufälligen Verkehrsmustern angenommen [GWT10]. Tatsächlich aufgenommen wird diese Energie allerdings nicht durch die Kanäle, son-

dern durch die Treiber, welche eine Bitübertragung veranlassen. Für Router ergibt sich aus [YBM04] näherungsweise ein Wert von 1,5 pJ je Bitübergang. Dieser Wert wird als gleichmäßig auf die Eingangs- und Ausgangsmodule verteilt angenommen, während nur wenige fJ auf die Verdrahtungsmatrix entfallen. Bezuglich der FIFOs wird angenommen, dass diese in die Eingangsmodule der einzelnen Ports integriert sind. Daher existiert bezüglich E_{Trans} kein separater Wert für diese. Der Wert von 1,5 pJ gilt für Router mit einer Flit-Breite von 64 Bit und einer FIFO-Tiefe von 2 Einträgen. Darüber hinaus kommen WHS und ein lastorientiertes XY-Routing zum Einsatz. Für IPCs ist aufgrund der Vielzahl und Heterogenität möglicher IPCs bezüglich E_{Trans} kein einzelner repräsentativer Wert bestimmbar, sodass als grobe Abschätzung 20 pJ angenommen werden. Der gewählte Wert ist wissenschaftlich nicht fundiert und dient vielmehr dazu, der schwankenden Wärmegenerierung durch variierende Lastprofile der IPCs sowie dem Größenverhältnis von IPC zu Router Rechnung zu tragen. Die Werte für E_{Trans} sind in Tabelle 3.2 zusammengefasst.

P_{Leck} repräsentiert die durch Leckströme hervorgerufene statische Leistungsaufnahme (siehe Gleichung 2.1) und ist nur für aktive Komponenten von Bedeutung. Dies umfasst neben den IPCs auch die Router, jedoch nicht die Übertragungskanäle. Für IPCs wurde P_{Leck} ausgehend von einem IBM PowerPC 405 [Pow13] bei 55 °C Betriebstemperatur und 1,2 V Betriebsspannung mit 100 mW festgelegt, da eine solche Komponente für eine Integration in ein NoC potentiell geeignet wäre. Für Router wird P_{Leck} für die einzelnen Ports separat angegeben. Der Wert ist dabei abhängig von der gewählten Flitbreite, der Lage des konkreten Ein- oder Ausgangsmoduls sowie der Größe des Pufferspeichers. Die Werte entstammen der in [C.11] durchgeführten Synthese eines NoC-Routers unter Verwendung der 65 nm-Technologie und gelten für eine Flitbreite von 64 Bit und eine FIFO-Größe von 8 Flit unter Verwendung von XY-Routing. Die Lageabhängigkeit der Leckströme ergibt sich aus der Optimierung der Ein- und Ausgangsmodule für das XY-Routing. Dieses macht aufgrund von Restriktionen bezüglich der Pfadwahl unterschiedliche Logikteile der einzelnen Module überflüssig. Die Werte für P_{Leck} sind ebenfalls in Tabelle 3.2 aufgeführt.

Modellierungsgranularität Das Temperaturmodell ist derart gestaltet, dass sich Größe und Anzahl der das System repräsentierenden Kacheln beeinflussen lassen. Die somit einstellbare Modellierungsgranularität hat maßgeblichen Einfluss auf die Dauer und die Genauigkeit der Temperatursimulation. Einerseits sorgt eine höhere Auflösung für ein filigraneres RC-Netz und somit für eine exaktere Modellierung des Wärmeflusses. Allerdings erhöhen sich dadurch auch der Simulations- und der Zeitaufwand. Andererseits hat eine niedrige Auflösung zwar eine schnellere Simulation, aber auch ungenauere und möglicherweise unbrauchbare Resultate zur Folge.

Tabelle 3.2: Energieaufnahme je Bitübergang E_{Trans} sowie statische Leistungsaufnahme P_{Leck} für einzelne Router-Komponenten, den IPC und den Übertragungskanal als Eingangsdaten für das Temperaturmodell (Flitbreite: 64 Bit, FIFO-Größe: 8 Flits)

	Komponente	E_{Trans}	P_{Leck}
Router	Eingangsmodul Nord und Süd	je Modul 1,5 pJ	36,26 nW
	Eingangsmodul Ost und West		64,92 nW
	Eingangsmodul IPC		64,92 nW
	Ausgangsmodul Nord und Süd		153,44 nW
	Ausgangsmodul Ost und West		42,48 nW
	Ausgangsmodul IPC		153,44 nW
	FIFO		1813,48 nW
IPC		20 pJ	100 mW
Kanal		11,62 fJ	/

Die Anzahl der Stromquellen bleibt hingegen unabhängig von der gewählten Granularität konstant und beschränkt sich auf eine Stromquelle je Komponente. Auch hier wäre eine feinere Granularität möglich, sodass die einzelnen Systemkomponenten weiter in Module mit dedizierten Stromquellen unterteilt werden könnten. Für einen einzelnen Router würde sich beispielsweise die Anzahl der Stromquellen auf fünf erhöhen. Der Vorteil wäre eine exaktere Modellierung des Ursprungs der in das System abgestrahlten Wärme. Der Nachteil wäre ein ungleich höherer Aufwand sowohl für die Aktivitätsüberwachung als auch die Temperatursimulation, der mit steigender NoC-Größe voraussichtlich stark zunimmt. Aus diesem Grund wird auf eine diesbezügliche Untersuchung verzichtet, da nicht eine möglichst exakte Simulation des Temperaturverlaufs, sondern eine effiziente und hinreichend exakte Temperaturmodellierung für ein laufzeitfähiges proaktives Systemmanagement Gegenstand dieser Arbeit ist. Für die Nutzung eines feiner aufgelösten Stromquellennetzes wäre zwar auch die gleichmäßige Verteilung des Gesamtstromes auf die einzelnen Stromquellen der Systemkomponenten denkbar. Dieser Ansatz läuft jedoch der Intention einer exakteren Modellierung des Wärmeursprungs zuwider und erhöht dennoch den Modellierungsaufwand.

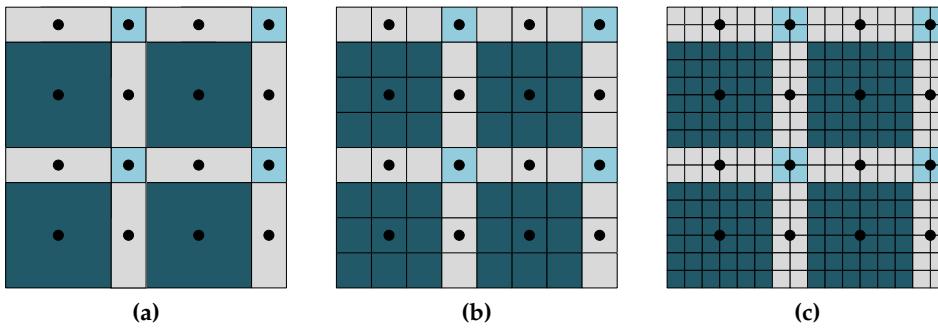


Abbildung 3.3: Genauigkeitsstufen der Abbildung eines 2×2 NoC auf ein äquivalentes RC-Netz: (a) Komponentenweise (**Auflösung R0**), (b) Router als kleinste Einheit (**Auflösung R1**), (c) Hälften der Router-Kantenlänge als kleinste Einheit (**Auflösung R2**)

Das Modell unterstützt die in Abbildung 3.3 dargestellten Auflösungen R0 bis R2. Für die Auflösung R0 wird jede Komponente durch eine einzelne RC-Kachel samt zugehöriger Stromquelle modelliert (siehe Abbildung 3.3(a)). Diese eher grobe Auflösung eignet sich vor allem für eine schnelle, aber vermutlich auch ungenaue Temperatursimulation. Es werden lediglich so viele Kacheln und Stromquellen benötigt, wie Komponenten vorhanden sind. Für R1 wird hingegen die Kantenlänge eines Routers als Maß für die Kantenlänge der RC-Kacheln verwendet. Je nach Dimensionierung der IPCs sind damit unterschiedlich viele Kacheln nötig, um diese und die Übertragungskanäle zu modellieren (siehe Abbildung 3.3(b)). Das am feinsten aufgelöste und somit akkurateste und aufwändigste Temperaturmodell ist exemplarisch in Abbildung 3.3(c) dargestellt. Hierbei entspricht die Kantenlänge der RC-Kacheln der Hälfte der Kantenlänge eines Routers. Für R1 vervierfacht sich für das gewählte Beispiel die Anzahl der RC-Kacheln im Vergleich zu R0. Für R2 ist der Aufwand im Vergleich zu R0 bereits sechzehn mal so hoch. Diese Relationen bleiben für wachsende NoC-Dimensionen erhalten, sodass ab einer bestimmten Größe der Zeitaufwand für die Berechnungen vermutlich inakzeptabel wird.

Geometrische Modellierung und weitere Parameter Neben dem eigentlichen Chip wird zusätzlich das sogenannte Cooling Package modelliert (siehe Abbildung 3.4(a)). Dieses besteht aus dem als Spreader bezeichneten Wärmeverteiler und dem Kühlkörper. Damit wird, dem Ansatz in [SSH⁺03] folgend, eine auch in der Praxis angewandte Technik zur Abführung der erzeugten Wärme integriert. Auf diese Weise wird ein realistischeres Verhalten des Modells erzeugt, da der Großteil der Wärme entlang des vertikalen Pfades über Spreader und Kühlkörper in die Umgebung abgeführt wird. Der Spreader sowie der Kühlkörper werden ebenfalls mithilfe von RC-Gliedern modelliert. Zu diesem Zweck werden beide Komponenten in fünf RC-Kacheln eingeteilt (siehe Abbildung 3.4(b)). Dabei werden allerdings keine Stromquellen platziert, da das Package

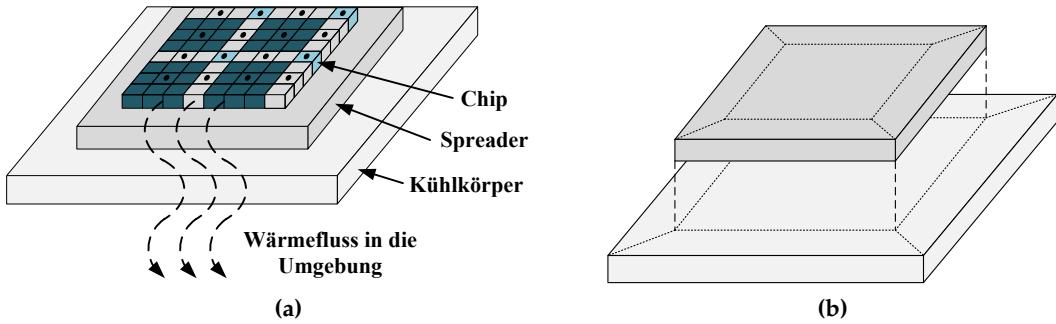


Abbildung 3.4: Vollständig modellierter Chip: (a) Wärmeabstrahlung über Wärmeverteiler (Spreader) und Kühlkörper (b) Einteilung von Spreader und Kühlkörper in jeweils fünf RC-Kacheln

selbst keine Wärme erzeugt, sondern diese nur an die Umgebung abführt. Die horizontale Wärmeabstrahlung wurde aufgrund der Ummantelung des Chips und der damit einhergehenden geringen thermischen Leitfähigkeit zugunsten einer Modellvereinfachung vernachlässigt. Angenommen wird ein Keramikmantel, im Allgemeinen wird jedoch auf kostengünstigere Plastikummantelungen zurückgegriffen. Im Gegensatz zum eigentlich aus Silizium bestehenden Chip werden Spreader und Kühlkörper als Kupferkörper modelliert. Dieser Unterschied schlägt sich in den thermischen Kapazitäten und Widerständen der jeweiligen Komponenten nieder (siehe Gleichungen 3.1 und 3.2).

Sämtliche Komponenten werden durch ihre geometrischen Dimensionierungen beeinflusst. Hierzu zählen neben der Kantenlänge von Routern und IPCs auch die Dicke des Chips, des Spreaders und des Kühlkörpers. Die Kantenlänge eines Routers wird für das Modell mit $141\text{ }\mu\text{m}$ angenommen, während ein IPC eine Kantenlänge von $1,85\text{ mm}$ aufweist. Diese Werte wurden ausgehend von den im SPICE-basierten Vergleichsmodell festgelegten Abmessungen gewählt, um die Vergleichbarkeit zu gewährleisten. Für ein 2×2 NoC ergibt sich damit zum Beispiel eine Chipfläche von etwa $4 \times 4\text{ mm}$. Die Dicke von Chip, Spreader und Kühlkörper wird nach [SSH⁺03] mit $0,6\text{ mm}$, 1 mm und $6,8\text{ mm}$ festgelegt. Die Kantenlänge des Spreaders ergibt sich aus der anderthalbfachen Kantenlänge des Chips. Für ein 2×2 NoC ergibt dies 6 mm . Die Ausmaße des Kühlkörpers ergeben sich wiederum aus der doppelten Kantenlänge des Spreaders. Für das Beispiel ergibt dies 12 mm . Dieser Aufbau gewährleistet, dass sich die Dimensionen von Spreader und Kühlkörper der mit variierender NoC-Größe schwankenden Chip-Größe anpassen.

Neben geometrischen und architekturbezogenen Informationen sind auch Parameter für die Simulation des Temperaturprofils des betrachteten NoC-basierten Systems relevant. Dies betrifft unter anderem die Umgebungs- und die anfängliche Chiptemperatur, welche den Verlauf der Temperaturentwicklung stark beeinflussen. Zusätzlich können die eingangs eingeführte Simulationsperiode T_S sowie eine Abtastperiode T_A gewählt werden. T_S beeinflusst die Länge des simulierten Zeitabschnitts, nach welchem

die Spannungsabfälle aus der Temperaturberechnung ausgelesen werden. Darüber hinaus werden nach Ablauf von T_S die Aktivitätsstatistiken für die vergangene Periode zur Berechnung des in das RC-Netz einzuspeisenden Stroms ausgewertet. T_S hat damit maßgeblichen Einfluss auf Leistungsfähigkeit und Vorhersagefähigkeit des Temperaturmodells. Eine sehr kurze Periode geht mit vielen, möglicherweise unnötigen Berechnungs- und Auswertungsphasen einher, während eine sehr lange Periode aufgetretene Aktivitäten unter Umständen erst sehr spät in die Temperaturberechnung einfließen lässt. T_A hingegen stellt die interne Abtastperiode der SystemC-eigenen AMS-Bibliothek dar und beeinflusst die Häufigkeit, mit der diverse Funktionen auf Basis eingehender Abtastwerte ausgeführt werden. Im Einzelnen bestimmt T_A die Häufigkeit, mit der modellinterne Funktionen für die Temperaturberechnung aufgerufen werden und beeinflusst damit den Verlauf des ausgegebenen Temperaturprofils sowie den Berechnungsaufwand.

Ablauf der Simulation

Die Simulation des Temperaturverhaltens gliedert sich in Start-, Setup- und Simulationsphase. Der gesamte Ablauf der parallelen Simulation von funktionellem Verhalten und dem daraus resultierenden Temperaturverlauf ist in Abbildung 3.5 dargestellt.

Start- und Setup-Phase Zunächst werden in der Startphase die nötigen Parameter für die funktionale Simulation und für die Erstellung des Temperaturmodells ausgewertet. Anschließend wird in der Setup-Phase zunächst das NoC-basierte System instanziert (siehe ①). Die integrierten IPCs werden lediglich als Sende- und Empfangseinheiten ohne jegliche weitere Funktionalität modelliert. Im Anschluss daran wird auf Basis der topologischen Eigenschaften des NoC sowie weiterer geometrischer Eigenschaften das Temperaturmodell erstellt. Dies beinhaltet zunächst die Instanziierung des äquivalenten RC-Netzes (siehe ②), wobei die nötigen Beschreibungen der elektrischen Primitive unter Verwendung des ELN-Berechnungsmodells aus der SystemC AMS-Bibliothek extrahiert werden. Hierzu zählen unter anderem Widerstände, Kapazitäten und Stromquellen. Weiterhin umfasst das Erstellen des Temperaturmodells die Verbreitung der gewählten internen Abtastperiode T_A im RC-Netz und dazugehörige Berechnungen (siehe ③).

$$u_{p,n}(t) = i_{p,n}(t) \cdot R \quad (3.5)$$

$$i_{p,n}(t) = C \cdot \frac{d(u_{p,n}(t) + \frac{q_0}{C})}{dt} \quad (3.6)$$

Der letzte Schritt zur Erstellung des Temperaturmodells ist das Aufstellen eines Differentialgleichungssystems (DGS) basierend auf den ELN-Modulen des RC-Netzes und eine Überprüfung der Lösbarkeit desselben (siehe ④). Die entsprechenden mathema-

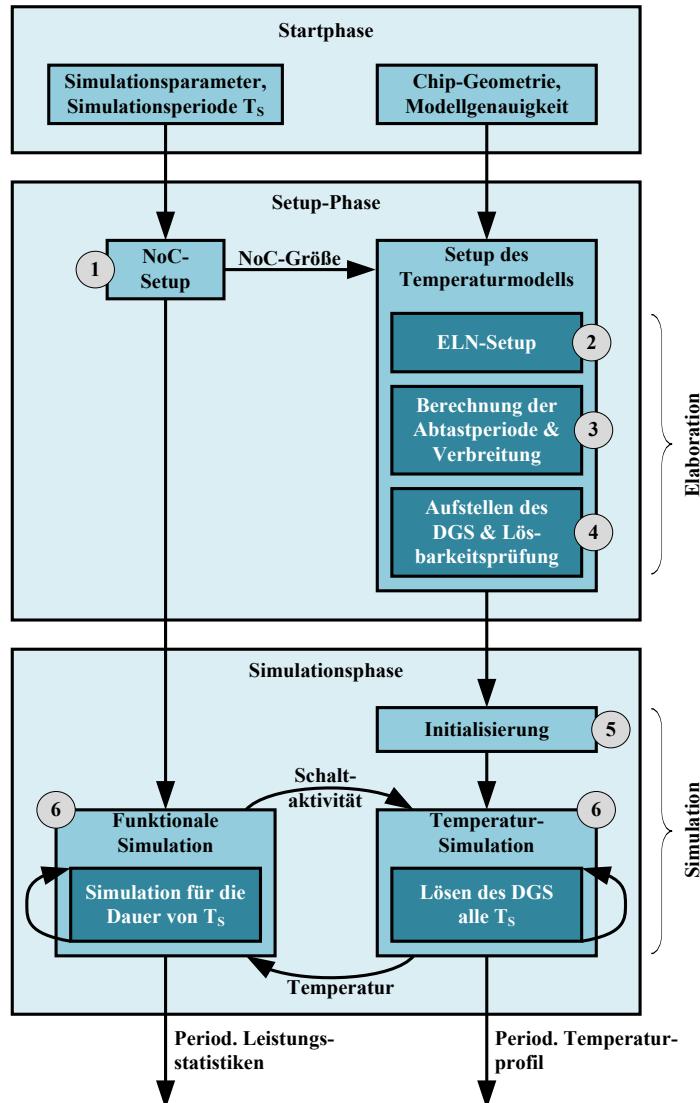


Abbildung 3.5: Ablauf der Simulation des Temperaturverlaufs

tischen Beschreibungen für elektrische Widerstände und Kapazitäten liefern die Gleichungen 3.5 und 3.6. Mithilfe solcher Gleichungen wird das gesamte Gleichungssystem entsprechend der Kirchhoffsschen Regeln für Ströme und Spannungen aufgestellt. In Abbildung 3.6 ist beispielhaft ein einfaches RC-Netz dargestellt. Das resultierende DGS für die Spannungsknoten u_a und u_b ergibt sich aus den Gleichungen unter 3.7. Dieses wird innerhalb des SystemC-Modells mithilfe numerischer Verfahren wie beispielsweise dem Runge-Kutta-Verfahren oder dem impliziten Euler-Verfahren gelöst. Die periodische Lösung des DGS liefert während des Simulationsverlaufs die zu einer Temperatur äquivalenten Spannungswerte. Die Schritte ② bis ④ werden auch als ELN-Elaboration zusammengefasst. Je nach gewählter NoC-Größe und Auflösung des Temperaturmodells

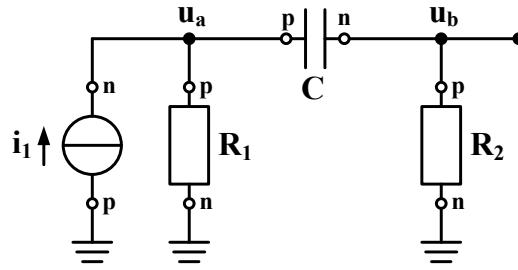


Abbildung 3.6: Beispiel für ein mithilfe von ELN-Primitiven modellierbares elektrisches Netzwerk [AMS13]

nimmt die Setup-Phase unterschiedlich viel Zeit in Anspruch, da besonders das Aufstellen und Überprüfen des DGS sehr aufwändig ist.

Simulationsphase Nachdem sowohl der funktionale als auch der temperaturbezogene Teil der Simulationsumgebung konfiguriert wurden, beginnt die eigentliche Simulationsphase. Zunächst werden ausgehend von den Simulationsparametern für alle ELN-Primitive die Ausgangskonditionen hergestellt (siehe ⑤). Im Anschluss daran finden parallel die Simulation des funktionalen Verhaltens und die dazugehörige Berechnung des Temperaturverlaufs des simulierten NoC-basierten Systems statt (siehe ⑥). Dabei wird die gesamte Simulation schrittweise für die Dauer von T_S vorangetrieben bis die Gesamtdauer erreicht ist. Zu den Abtastzeitpunkten wird das DGS numerisch gelöst. Die ermittelten, einer Temperaturänderung entsprechenden Spannungswerte werden dem funktionalen Teil der Simulationsumgebung für ein temperaturorientiertes Management übergeben (siehe Abschnitte 4.2 und 4.3). Des Weiteren werden die, auf Basis des in der vorigen Periode aufgezeichneten Aktivitätsprofils, nach Gleichung 3.4 berechneten aktualisierten Ströme in das RC-Netz eingespeist. Die Aufzeichnung des Aktivitätsprofils erfolgt im funktionalen Teil der Simulation.

$$DGS_{a,b} \begin{cases} -i_1 + \frac{u_a}{R_1} + C \cdot \frac{d(u_{a,b} + \frac{q_0}{C})}{dt} = 0 \\ \frac{u_b}{R_2} - C \cdot \frac{d(u_{a,b} + \frac{q_0}{C})}{dt} = 0 \end{cases} \quad (3.7)$$

Dies verdeutlicht erneut die Bedeutsamkeit der richtigen Wahl der Simulationsperiode, da durch einen zu großen Wert die am Anfang einer Periode aufgetretenen Aktivitäten erst sehr spät in die Berechnungen einfließen können. Umgekehrt sorgt eine sehr kurze Periode unter Umständen für unnötig viele Berechnungsschritte. Die Unterbrechung der Simulation hat im Übrigen keinerlei Einfluss auf den Simulationsverlauf oder die erzielten Ergebnisse, da entsprechende SystemC-Methoden für eine einheitliche Unterbrechung und Wiederaufnahme aller Abläufe ohne Datenverlust Sorge tragen [Sys]. Die Schritte ⑤ und ⑥ werden als ELN-Time-Domain-Simulation zusammengefasst.

Ein essentieller Vorteil der in diesem Abschnitt vorgestellten Simulationsumgebung ist die Kombination der Vorteile einer SPICE-basierten Simulation der thermischen Parameter eines NoC-basierten Systems und einer SystemC-gestützten funktionalen Simulation dieses Systems. Einerseits bietet das in SystemC AMS integrierte ELN-Berechnungsmodell die gleichen elektrischen Primitive, die auch SPICE auszeichnen. Dies stellt eine akkurate Modellierung des thermischen Verhaltens auf dem Niveau einer SPICE-Simulation sicher. Andererseits wird die statische Natur einer SPICE-basierten Modellierung durch einen relativ frei steuerbaren und pausierbaren zeitlichen Simulationsablauf umgangen. Damit lassen sich auch während einer laufenden Simulation neue Stimuli einspeisen oder Statistiken extrahieren.

Weiterhin ist die Umgebung unabhängig von externen Werkzeugen wie Wattch oder Orion [BTM00, WZPM02], welche üblicherweise die für eine Simulation des Temperaturprofils benötigten Werte für die Leistungsaufnahme bereitstellen. Die vorgestellte Simulationsumgebung greift zwar ebenfalls auf Durchschnittswerte für die Leistungsaufnahme einzelner Systemkomponenten zurück und kombiniert diese mit aufgezeichneten Aktivitäten. Jedoch geschieht dies ohne äußere Eingriffe und parallel zur Simulation des funktionalen Verhaltens des betrachteten Systems, sodass bereits während der Simulation die Temperaturentwicklung berechnet werden kann. Die genannten externen Werkzeuge hingegen setzen die Verfügbarkeit der Statistiken bezüglich der Leistungsaufnahme für den gesamten Simulationszeitraum voraus. Dies erhöht zwar im Vergleich zur vorgestellten Umgebung die Modellierungsgenauigkeit, schließt jedoch einen parallelen Ablauf von funktionaler Simulation und Temperaturberechnung von vornherein aus.

3.2 Bewertung des Modells

In diesem Abschnitt wird das vorgestellte Temperaturmodell bezüglich seiner Leistungsfähigkeit sowie seiner Korrektheit bewertet. Um quantitative Vergleiche anstellen zu können, wird ein identisches SPICE-basiertes Modell herangezogen [TCST10]. Dieses wurde am Institut für Angewandte Mikroelektronik und Datentechnik entwickelt und wird in ähnlicher Form in [LCN⁺09] genutzt. Es nutzt ebenfalls den in Abschnitt 3.1.1 erläuterten Dualismus elektrischer und thermischer Parameter, basiert auf der Simulation einer Netzliste und weist als einzigen Unterschied das Fehlen der Auflösung R0 auf. Auf diese wurde zugunsten eines geringeren Implementierungsaufwands und aufgrund der nicht vollständig dem Modell des regulären RC-Netzes entsprechenden Struktur verzichtet.

Die Netzliste wird zunächst auf Basis der geometrischen Daten des zugrunde liegenden Systems sowie dessen Wärmequellen generiert. Als weitere Eingangswerte werden die Aktivitätsprofile sämtlicher Systemkomponenten über die gesamte zu simulierende Zeit benötigt. Mit deren Hilfe wird die durch die einzelnen Systemkomponenten

Tabelle 3.3: Simulationsparameter für die Verifikation des Temperaturmodells NoC-basierter Systeme

	Parameter	Wert
Architektur	Topologie	Gitternetz
	Adressierung	Geografisch
	Switching	WHS
	Routing	XYR
	Arbitrierung	Round-Robin
	Flitbreite	64 Bit
Router	FIFO-Größe	8 Flits
	Taktfrequenz	1 GHz
	Paketgröße	64 - 2000 Flits
	Verkehrsmuster	zufällig, gleichverteilt
Simulation	Simulationsdauer	1 s
	Anfängl. Chiptemperatur	60 °C
	Umgebungstemperatur	45 °C
Temperaturmodell		

erzeugte Wärme unter Verwendung äquivalenter Stromquellen nachgebildet. Die wichtigsten Eckpunkte der im Folgenden durchgeführten Simulationen sind in Tabelle 3.3 aufgelistet. Die anfängliche Chiptemperatur und die Umgebungstemperatur betragen 60 °C [SPKJ04] beziehungsweise 45 °C [SSH⁺03]. Das Adressierungsschema ist geografischer Natur. Dies bedeutet, dass sich die Adressen der einzelnen IPCs und der dazugehörigen Router in eine X- und eine Y-Komponente aufteilen. Diesen Komponenten werden von unten links nach oben rechts wachsende Zahlen zugeordnet. In einem 2×2 NoC (siehe beispielsweise Abbildung 3.2) besitzt der IPC links unten die Koordinaten (0,0), während der IPC rechts oben durch die Koordinaten (3,3) gekennzeichnet ist.

Ein Aspekt, der mit Sicherheit für Abweichungen zwischen den beiden Modellen sorgt und bei der Bewertung berücksichtigt werden muss, ist die unterschiedliche Realisierung der äquivalenten Stromquellen. Während die SPICE-basierte Implementierung stückweise lineare Quellen (Piecewise Linear Source (PWL)) nutzt, werden die AMS-Quellen zeitkontinuierlich modelliert. Der Strom einer PWL wird als lineare Funktion dargestellt, welche sich aus den gegebenen Wertepaaren von Zeitpunkt und Strom ergibt [HSP]. Für AMS-Quellen werden solche diskreten Wertepaare genutzt, um einen rea-

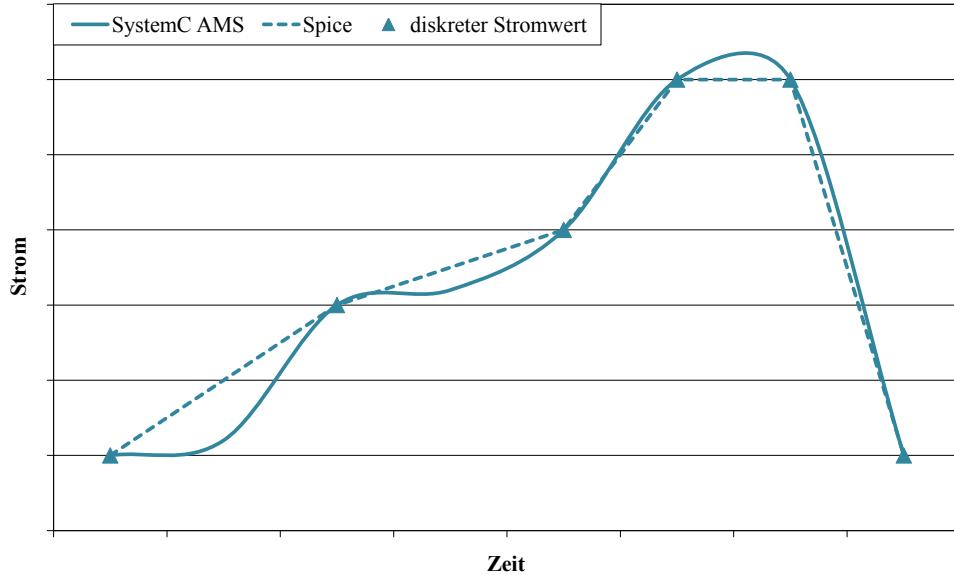


Abbildung 3.7: Modellierung einer Stromquelle anhand diskreter Wertepaare von Zeit und Strom als stückweise lineare Funktion mit SPICE und als zeitkontinuierliche Funktion mit SystemC AMS

listischeren, zeitkontinuierlichen Stromverlauf zu erzeugen [AMS]. Dieser Unterschied in der Modellierung der Stromquellen ist beispielhaft in Abbildung 3.7 dargestellt. So ergeben sich für identische diskrete Wertepaare von Zeit und Strom mit voranschreitender Zeit zwangsweise verschiedene Werte für die Wärmemenge. Die insgesamt erzeugte Wärmemenge wird durch die Fläche repräsentiert, welche von der jeweiligen Kurve eingeschlossen wird. Erst ein neues Wertepaar bildet einen Synchronisationspunkt, an welchem die Quellen abgeglichen werden können. Die Wahl der Simulationsperiode T_S (siehe Abschnitt 3.1.3) hat dabei einen nicht zu vernachlässigenden Einfluss, da sie die Anzahl der diskreten Wertepaare beeinflusst. Dies führt unvermeidlich zu unterschiedlichen Wärmeprofilen. Daher ist eine exakte Annäherung des AMS-basierten Modells an das Referenzmodell nicht möglich, aber auch nicht zielführend. Einerseits ist nicht eine exakte Übereinstimmung absoluter Temperaturwerte, sondern ein möglichst identischer Kurvenverlauf der Temperaturprofile entscheidend. Andererseits erscheint ein zeitkontinuierlicher Verlauf des eingespeisten Stroms ohnehin realitätsnäher.

3.2.1 Zeitverhalten

In diesem Abschnitt wird die Leistungsfähigkeit des AMS-basierten Temperaturmodells untersucht. Die Leistungsfähigkeit bezieht sich dabei auf die Geschwindigkeit der Berechnung des Temperaturprofils für eine gegebene, zu simulierende Zeitspanne. Dabei

ist nicht nur ein Leistungsvorsprung des AMS-basierten Systems vor dem Referenzmodell wünschenswert. Idealerweise wird die Berechnung des Temperaturprofils des Gesamtsystems für einen bestimmten Zeitraum vor Ablauf dieses Zeitraums vollendet. So sollte beispielsweise die Kalkulation des Temperaturverhaltens eines NoC-basierten Systems über eine Simulationsdauer von 5 s im günstigsten Fall deutlich weniger Zeit in Anspruch nehmen als 5 s. Eine solche Leistungsfähigkeit ist im Hinblick auf den Einsatz des Temperaturmodells für die in Kapitel 4 vorgestellten Konzepte für ein proaktives temperaturorientiertes Management NoC-basierter Systeme von großer Bedeutung, da auf diese Weise Temperaturentwicklungen in Echtzeit abgeschätzt werden könnten.

Genauigkeitsstufen Da sowohl das implementierte Temperaturmodell als auch das SPICE-basierte Referenzmodell verschiedene Genauigkeitsstufen unterstützen, werden zunächst diese bezüglich ihrer Auswirkung auf die Leistungsfähigkeit der Modelle untersucht. Dabei ist zu beachten, dass der in Abbildung 3.5 illustrierte Ablauf der eigentlichen AMS-basierten Temperatursimulation lediglich in der sechsten Phase stattfindet. Um möglichst genaue Aussagen über den zeitlichen Aufwand zu erhalten, wird die funktionale Simulation in diesem Abschnitt durch im Vorfeld generierte Aktivitätsstatistiken ersetzt. Andernfalls würde der entstehende Mehraufwand die Ergebnisse verfälschen. Da aber die in Abschnitt 3.1.3 erläuterte Phase der Elaboration des Temperaturmodells unter Umständen viel Zeit in Anspruch nimmt, muss auch diese in den folgenden Betrachtungen berücksichtigt werden. Insbesondere für feine Auflösungen ist mit einem erhöhten Zeitaufwand zu rechnen, da die Komplexität des zu lösenden DGS mit der Anzahl der RC-Kacheln schnell zunimmt. Des Weiteren muss die insgesamt benötigte Zeit betrachtet werden, da die SPICE-basierte Simulation sich nicht explizit in identische Phasen einteilen lässt. Ein ähnlicher interner Ablauf ist allerdings anzunehmen.

In Tabelle 3.4 sind die Ergebnisse für die Dauer der Temperatursimulation eines 2×2 NoC dargestellt. Darüber hinaus wird die Anzahl der für die Modellierung nötigen RC-Kacheln aufgeführt. Die Menge der Kacheln hat maßgeblichen Einfluss auf den zeitlichen Aufwand, da die resultierende Anzahl der verwendeten elektrischen Primitive die Komplexität des DGS bestimmt. Dies gilt insbesondere für die Elaborationsphase, also das Aufstellen und die Lösbarkeitsprüfung des DGS. Aufgrund sehr langer zu erwartender Berechnungszeiten beschränkt sich die Untersuchung daher auf die Betrachtung eines 2×2 NoC, um etwaige Trends zu identifizieren.

Wie erkennbar ist, nehmen sowohl der Zeitaufwand für die Elaboration als auch die Simulation mit feiner werdender Auflösung für das AMS-Modell zu. Während die Simulationsdauer mit steigender Anzahl der RC-Kacheln relativ moderat von etwa 350 ms auf über 6 s zunimmt, steigt der Aufwand für die Elaboration deutlich stärker von 9 ms auf über 85 Minuten. Mit steigender Auflösung nimmt damit die Gesamtdauer der Tempe-

ratursimulation unverhältnismäßig stark zu, sodass inakzeptable Wartezeiten auftreten. Dies trifft allerdings auch auf das SPICE-basierte Modell zu. Die Erhöhung der Anzahl der RC-Kacheln für dieses führt zu einer Gesamtdauer von fast 55 Minuten für die höchste Auflösung. Dieser Wert ist zwar deutlich geringer als der entsprechende Wert für das AMS-Modell, bleibt aber für eine effiziente Simulation nach wie vor unbrauchbar. Dies gilt ebenfalls für R1, wobei das AMS-basierte Modell sein SPICE-Pendant um etwa 58 % unterbietet. Es muss davon ausgegangen werden, dass das SPICE-Modell einen ähnlichen internen Ablauf aufweist, welcher aus Elaboration und Simulation besteht.

Nach dem derzeitigen Implementierungsstand ist somit lediglich das AMS-Modell mit der Auflösung R0 geeignet, das angestrebte Ziel einer unter der simulierten Zeit liegenden Simulationsdauer zu erreichen. Zusätzlich ist für größere NoCs mit einem erhöhten Simulationsaufwand zu rechnen, sodass die Simulationsdauer für diese in Kombination mit höheren Auflösungen noch sehr viel schneller inakzeptable Werte erreicht. Eine eingehendere Untersuchung diesbezüglich findet an anderer Stelle statt. Allerdings würde das SPICE-basierte Modell möglicherweise ähnliche Ergebnisse für die Auflösung R0 erzielen. Die am Ende des Abschnitts 3.1.3 beschriebene statische Natur des SPICE-Modells stünde damit allerdings nach wie vor einem Einsatz zur Laufzeit im Wege.

Da die Phase der Elaboration nicht zur eigentlichen Temperatursimulation gehört (siehe Abbildung 3.5), aber unter Umständen dennoch enorm viel Zeit beansprucht, wäre es wünschenswert, diese gerade für höhere Auflösungen und größere Systeme umgehen zu können. Dies würde sich insbesondere dann anbieten, wenn mehrere identische Simulationen des gleichen NoC-basierten Systems durchgeführt werden oder eine laufende Simulation abgebrochen wird und zu einem späteren Zeitpunkt wieder aufgenommen werden soll. Für solche Fälle wäre eine erneute Elaboration theoretisch überflüssig, da das zugrunde liegende DGS lediglich von unveränderten, geometrischen und architekturebezogenen Parametern abhängig ist. Zudem ist die Phase der Elaboration klar von der Simulationsphase getrennt. Somit muss die Elaboration erst beendet werden, bevor die funktionale Simulation und der Simulation des Temperaturverhaltens parallel ablaufen können. Damit sagt der zeitliche Umfang der Elaborationsphase nur wenig über die Echtzeitfähigkeit und Leistungsfähigkeit des AMS-Modells aus.

Trotzdem wäre es beispielsweise denkbar, die Simulationsumgebung mit einer Bibliothek verschiedener prä-elaborierter NoC-basierter Systeme auszustatten, sodass lediglich das korrekte System geladen werden muss und die eigentliche Simulation umgehend beginnen kann. Für die Umsetzung dieses Ansatzes erscheinen speziell auf SystemC ausgerichtete „Save-and-Restore“-Techniken [MEB09] als besonders geeignet. Die Speicherung der Elaborationsergebnisse für eine spätere Wiederverwendung wird zudem bereits im SystemC Benutzerhandbuch [Sys] vorgeschlagen. Eine Implementierung konnte im Rahmen dieser Arbeit aufgrund der Komplexität dieses Themas nicht umge-

Tabelle 3.4: Zeitaufwand und Anzahl der nötigen RC-Kacheln für die Temperatursimulation eines 2×2 NoC über die Dauer von 1 s für verschiedene Auflösungen

	R0		R1		R2	
	AMS	SPICE	AMS	SPICE	AMS	SPICE
Elaboration [s]	0,009	/	87,035	/	5107,85	/
Simulation [s]	0,351	/	1,226	/	6,108	/
Gesamt [s]	$\approx 0,4$	/	≈ 88	209	≈ 5114	3292
RC-Kacheln	16	/	784	784	3136	3136

setzt werden, da hierfür weitreichende Anpassungen der SystemC-eigenen Bibliotheken notwendig wären.

Mit dem Ziel einer möglichst hohen Leistungsfähigkeit ist, ohne die Möglichkeit der Wiederverwendung der Elaborationsergebnisse, die Wahl der Auflösung R0 somit die praktikabelste Lösung. Inwiefern dies für größere NoC-basierte Systeme gilt und welche Auswirkungen auf die Korrektheit des Modells zu erwarten sind, wird im Folgenden untersucht.

NoC-Größe Der Einfluss der Größe des NoC-basierten Systems auf die benötigte Dauer für Elaboration und Simulation ist in Abbildung 3.8 dargestellt. Die Werte wurden für NoC-Größen von 2×2 bis 6×6 für eine simulierte Dauer von einer Sekunde unter Verwendung der Auflösung R0 aufgenommen. Wie erwartet, nimmt die Gesamtdauer mit steigender Systemgröße zu und folgt dabei einem exponentiellen Verlauf. Der Anteil der Simulationsphase an der gesamten Dauer ist mit 97 % für ein 2×2 NoC sehr hoch. Mit steigender NoC-Größe steigt zwar auch der zeitliche Aufwand für die Simulation, jedoch nimmt der Anteil der Elaboration an der Gesamtdauer sehr viel stärker zu. Die Simulation für ein 5×5 NoC beansprucht nur noch 74 % der gesamten Dauer und für ein 6×6 NoC entfallen lediglich 58 % der Gesamtdauer auf die Simulationsphase. Des Weiteren überschreitet die Gesamtdauer der Simulationen, außer für das 6×6 NoC mit 1007,7 ms, in keinem Fall die simulierte Dauer von einer Sekunde. Da vor der Fertigstellung der Elaboration die Temperatursimulation nicht stattfinden kann, ist diesbezüglich allerdings ohnehin nur die Dauer der Simulationsphase von Bedeutung. Diese bleibt auch für ein 6×6 NoC deutlich unterhalb einer Sekunde.

Für höhere Auflösungen steigt die Gesamtdauer wie erwartet drastisch an, sodass sowohl für das AMS-Modell als auch das SPICE-Modell bereits Werte von mehr als fünfzehn Minuten für ein 3×3 NoC erreicht werden. Dieser Verlauf setzt sich mit wachsen-

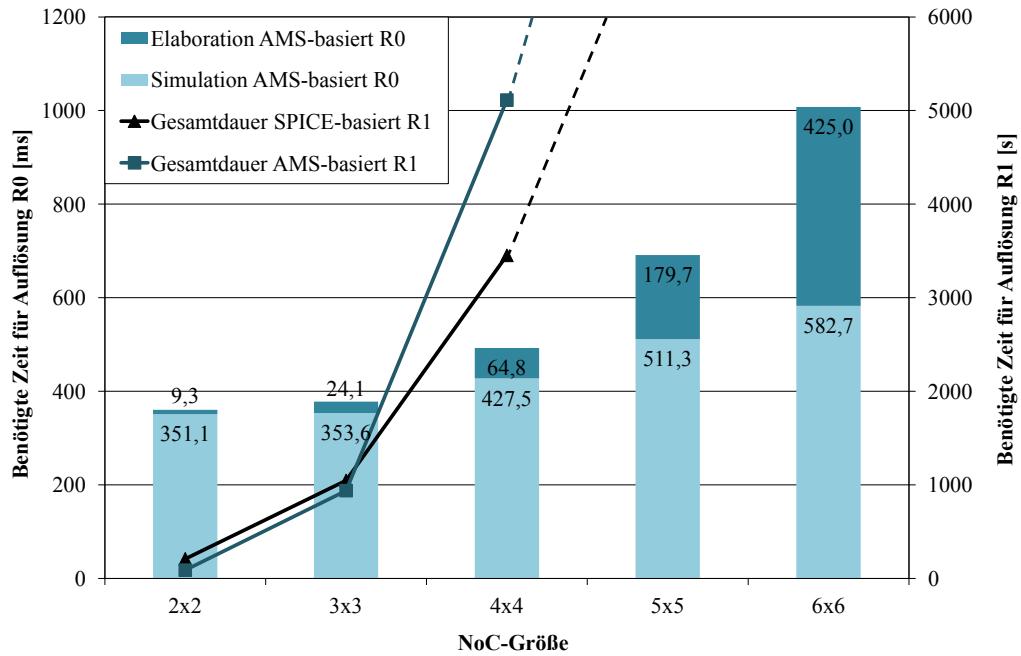


Abbildung 3.8: Zeitaufwand der Temperatursimulation in Abhängigkeit von der Größe des zugrunde liegenden Systems (Simulationsdauer: 1 s)

der NoC-Größe fort, wobei das SPICE-Modell ab einer Größe von 4×4 dem AMS-Modell deutlich überlegen ist. Dessen ungeachtet erzielen beide Modelle inakzeptable Resultate von mindestens einer Stunde Gesamtdauer. Dabei entfällt fast die gesamte Zeit auf die Elaborationsphase, während lediglich wenige Sekunden für die Simulation benötigt werden. Dennoch ist selbst der Zeitbedarf mehrerer Sekunden für die Simulation ein Ausschlusskriterium für höhere Auflösungen.

Dem Trend in Abbildung 3.8 folgend, kann selbst für größere NoCs das angestrebte Ziel, die Dauer der eigentlichen Simulationsphase unterhalb der simulierten Laufzeit zu halten, als gesichert gelten. Die Voraussetzung hierfür ist eine eher grob aufgelöste Modellierung des Temperaturverhaltens.

T_S und T_A Weitere Einflussfaktoren, welche die Leistungsfähigkeit des AMS-Modells beeinflussen, sind die in Abschnitt 3.1.3 eingeführten Simulationsparameter T_S und T_A . Die Auswirkung verschiedener Werte für diese Parameter ist in Abbildung 3.9 illustriert. Die Simulationsperiode T_S bestimmt die Häufigkeit, mit welcher die Aktivitätsstatistiken der funktionalen Simulation für die Einspeisung entsprechender Ströme in das RC-Netz ausgewertet und die Ergebnisse der Temperaturberechnungen abgerufen werden. Die Abtastperiode T_A bestimmt die Häufigkeit der Ausführung AMS-interner Funktionen. Sie beeinflusst die Anzahl der Berechnungsschritte und damit unter Umständen ebenfalls Genauigkeit und Leistungsfähigkeit des Temperaturmodells.

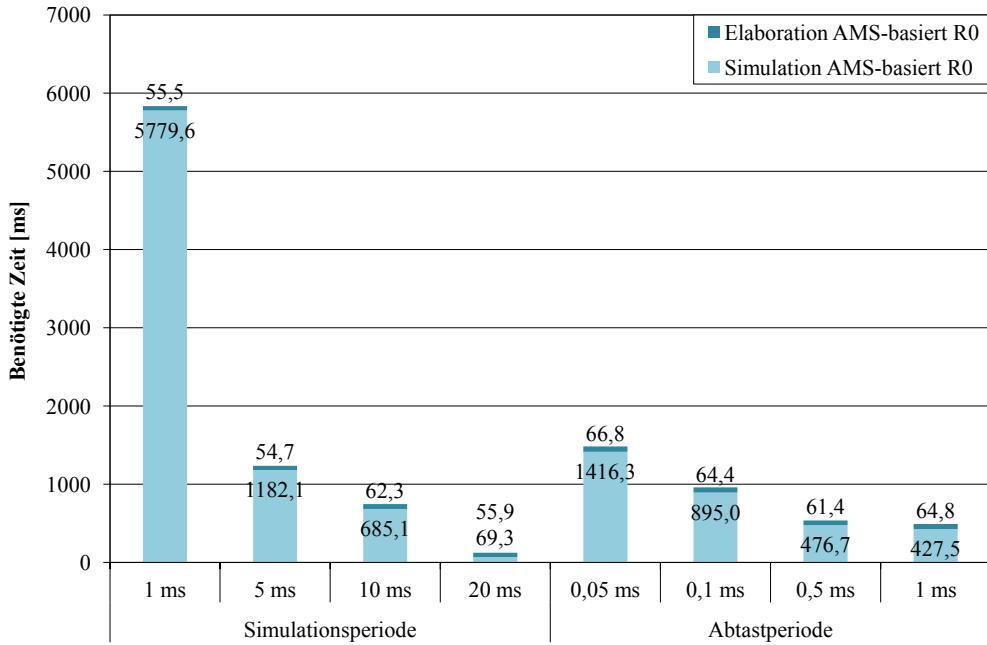


Abbildung 3.9: Zeitaufwand der Temperatursimulation in Abhängigkeit von der Simulationsperiode T_S und der Abtastperiode T_A für ein 4×4 NoC (Auflösung: R0, $T_A = 1$ ms für variable T_S , $T_S = 2$ ms für variable T_A)

Wie zu erkennen ist, wird durch unterschiedliche Werte für T_S und T_A lediglich die Dauer der Simulationsphase nennenswert beeinflusst. Besonders im Hinblick auf T_S sorgen kurze Perioden während der Simulationsphase für einen deutlichen Zuwachs der benötigten Dauer. Die Ursache ist ein sehr häufiges Pausieren der Simulation, um die Aktivitätsprofile abzurufen, die entsprechenden Ströme in das RC-Netz einzuspeisen und die resultierenden Auswirkungen zu berechnen. Aus einer leistungsorientierten Perspektive scheinen daher lange Simulationsperioden von Vorteil zu sein. Durch diese kann die Dauer der Simulationsphase auf weniger als 100 ms verkürzt werden. Ein potentieller Nachteil hoher Werte für T_S ist allerdings, dass eventuell am Anfang einer Periode auftretende Schaltaktivitäten erst nach Ablauf dieser Periode in das Temperaturmodell einfließen können. Eine proaktive Berechnung wäre damit unter Umständen unmöglich, da der entsprechende thermische Effekt in der Zwischenzeit bereits eingetreten sein könnte. Sehr kurze Werte hingegen bergen die Gefahr einer Überabtastung, sodass die Simulation unnötig häufig pausiert wird ohne nennenswerte Aktivitäten in das Modell einfließen lassen zu können. Um diesbezüglich eine konkrete Aussage treffen zu können, sind weitere Untersuchungen bezüglich des Einflusses von T_S auf die Korrektheit des Modells erforderlich. Diese werden in Abschnitt 3.2.2 durchgeführt. Voraussichtlich muss aber ein Kompromiss zwischen Aufwand und Genauigkeit gefunden werden, da die in einer Periode aufgezeichneten Aktivitätsstatistiken nicht nur von der

Periodenlänge sondern auch von der Anzahl und vom Kommunikationsverhalten der in das NoC integrierten IPCs abhängig sind.

Dieser Sachverhalt trifft für T_A jedoch nicht zu. Hier lassen sich klare Aussagen treffen, ob der durch kurze Abtastperioden verursachte Mehraufwand durch eine höhere Genauigkeit des Modells zu rechtfertigen ist. Hohe Werte bewirken eine Reduzierung der AMS-internen Berechnungsschritte für das Temperaturmodell und beeinflussen dadurch möglicherweise das Temperaturmodell in negativer Weise. Ein Abgleich mit den Resultaten in Abschnitt 3.2.2 schafft an dieser Stelle Klarheit. Um jedoch eine möglichst hohe Leistungsfähigkeit zu erzielen, sind hohe Werte für T_A zu wählen. Bei der Wahl dieser ist allerdings weiterhin zu beachten, dass der Wert für T_A den von T_S nicht überschreitet. Andernfalls werden Änderungen des Temperaturprofils weniger häufig berechnet als neue Aktivitätsstatistiken vorliegen.

3.2.2 Korrektheit des Modells

Neben dem zeitlichen Aufwand für die Simulation des Temperaturverlaufs spielen selbstverständlich auch die physikalische Korrektheit sowie ein nachvollziehbares Verhalten des Temperaturmodells eine wichtige Rolle. In diesem Abschnitt werden daher der Einfluss verschiedener Auflösungen sowie der Effekt verschiedener Werte für T_S und T_A auf allgemeine Temperaturverhältnisse im NoC sowie auf einzelne Komponenten untersucht. Das Ziel dabei ist jedoch nicht, die Parameter für eine möglichst genaue Nachbildung des SPICE-basierten Temperaturverlaufs zu ermitteln. Vielmehr soll eine hinreichend exakte Modellierung erreicht werden, um zukünftige Temperaturtrends zuverlässig voraussagen zu können. Somit steht die Korrektheit des Kurvenverlaufs des AMS-Modells bezogen auf das SPICE-Modell beziehungsweise die Relation zwischen den Modellen im Mittelpunkt. Der absolute Temperaturwert kann mithilfe anderer Parameter nachjustiert oder von Zeit zu Zeit an den physikalischen Temperatursensoren neu ausgerichtet werden.

Genauigkeitsstufen Der Einfluss der Auflösungen R0, R1 und R2 auf die Werte für die durchschnittliche Temperatur des NoC-basierten Systems über die Simulationszeit von einer Sekunde ist in Abbildung 3.10 dargestellt. Als Referenz dient der mithilfe des SPICE-Modells ermittelte Temperaturverlauf unter Verwendung der Auflösung R1. Die zugrunde liegende funktionale Simulation wurde in jeweils zwei aktive und zwei passive Phasen eingeteilt, um die Modellierung aktivitätsbedingter Temperaturanstiege und leerlaufbedingter Abkühlungen zu untersuchen. Wie erkennbar ist, weisen alle Kurvenverläufe des AMS-basierten Modells einen ähnlichen Verlauf auf. Die teils deutliche Abweichung der Auflösung R0 wird besonders in Phasen sichtbar, in denen ein Temperaturanstieg modelliert werden soll. Zusätzlich zeichnet sich eine zeitliche Verschiebung zwi-

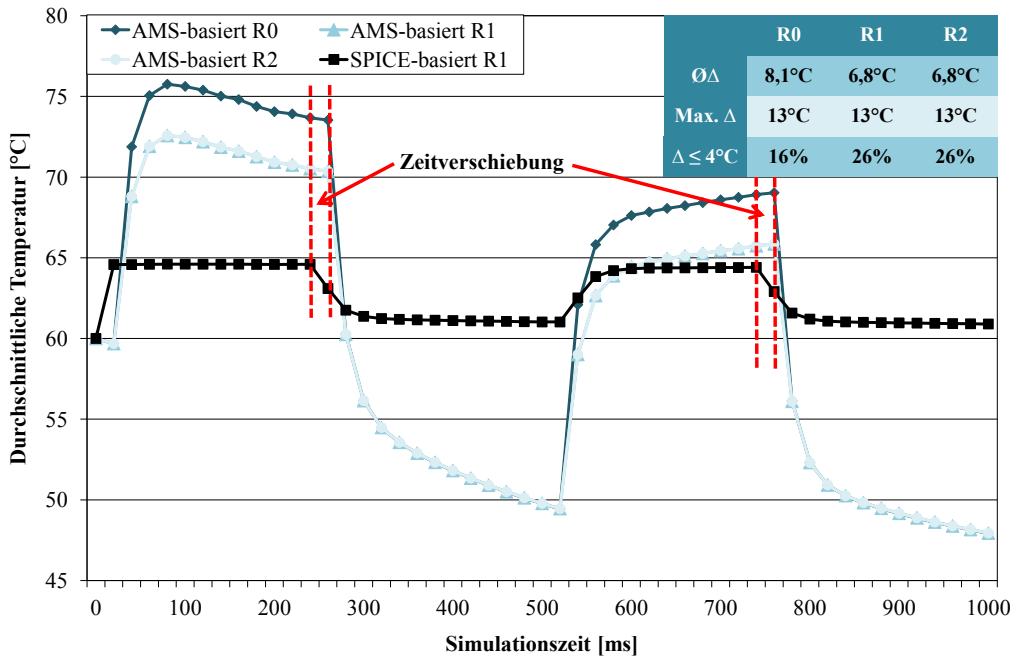


Abbildung 3.10: Durchschnittliche Temperatur aller Systemkomponenten eines 2×2 NoC unter Verwendung verschiedener Auflösungen

schen den Kurven des AMS-Modells und der Kurve des SPICE-Modells von etwa 20 ms ab. Die Ursache dafür liegt in der abweichenden Modellierung der Stromquellen mit SPICE und SystemC AMS (siehe Abbildung 3.7). Aufgrund des Algorithmus zur Bildung des zeitkontinuierlichen Signals ergeben sich unter Umständen verzögert einsetzende oder endende Phasen, in denen ein Strom eingespeist wird. Diese Abweichung lässt sich zwar nicht eliminieren, kann aber aufgrund ihrer konstanten Natur in die Berechnungen einbezogen werden. Weiterhin über- beziehungsweise unterschreiten die Kurven des AMS-basierten Modells den Verlauf des SPICE-basierten Modells in den entsprechenden Phasen. Dies weist auf Spielraum zur Verbesserung der Konfiguration des Modells hin. Beispielsweise würde eine Skalierung des aus den Aktivitätsprofilen berechneten Stromes an dieser Stelle Abhilfe schaffen. Der Verlauf des gesamten Temperaturprofils des modellierten Systems für einen Zeitraum von 500 ms ist in Abbildung 3.11 dargestellt.

Es wird deutlich, dass das Profil des AMS-basierten Modells (siehe Abbildungen 3.11(a) bis 3.11(d)) deutlich höhere Temperaturwerte und -schwankungen erzeugt als das SPICE-basierte Modell (siehe Abbildungen 3.11(e) bis 3.11(h)). Letzteres weist einen gleichmäßigeren Verlauf auf. Dies korrespondiert mit den in Abbildung 3.10 dargestellten Kurvenverläufen. Des Weiteren ist eine erhöhte Auflösung bis zu einem gewissen Grad in der Lage, die Abweichungen zu kompensieren. So reduziert R1 im Vergleich zu R0 die durchschnittliche Abweichung $\bar{\Delta}$ zum SPICE-basierten Modell um 1,3 %. Auch die Anzahl der Werte, welche eine Abweichung von weniger als 4°C aufweisen, kann

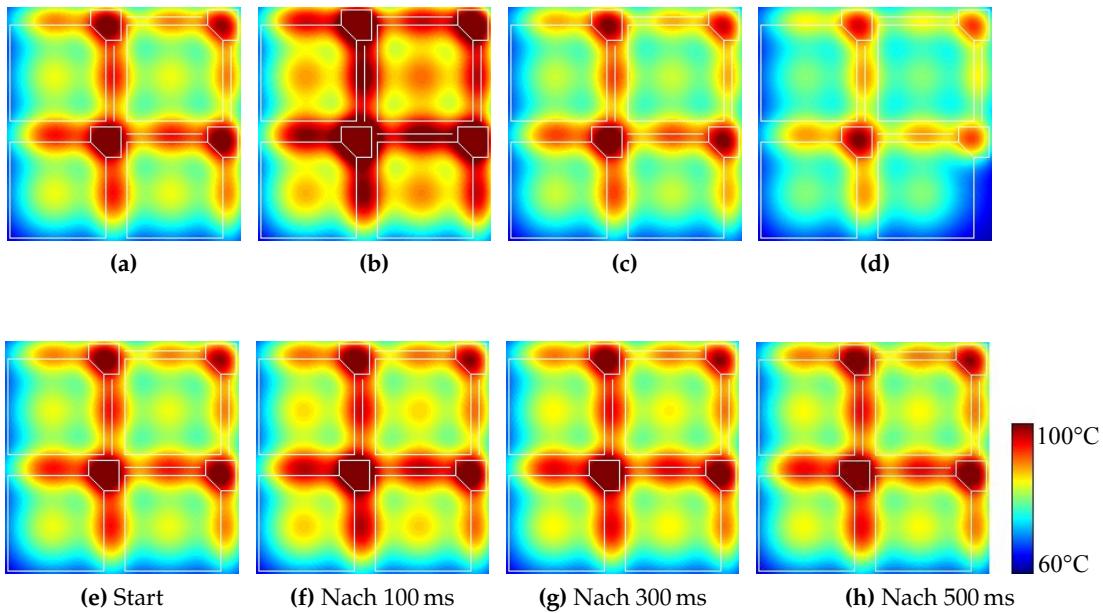


Abbildung 3.11: Modellierung des Temperaturprofils eines 2×2 NoC über einen Zeitraum von 500 ms: (a) bis (d) AMS-basiert (Auflösung: R0), (e) bis (h) mithilfe von SPICE (Auflösung: R1)

mit einer erhöhten Auflösung um 10 % gesteigert werden. Dieser Wert wurde gewählt, da er die absolute Genauigkeit eines auf einem Virtex5-FPGA verbauten Temperatursensors [GWWT12] angibt. Abweichungen unterhalb dieses Wertes liegen somit innerhalb der Messgenauigkeit gängiger Sensoren. Da eine weitere Erhöhung der Auflösung auf R2 keinerlei Verbesserungen mit sich bringt, wird diese nicht weiter betrachtet.

Die Analyse durchschnittlicher Temperaturwerte allein erlaubt keine umfassenden Aussagen über die Korrektheit des AMS-Modells. Aus diesem Grund wurde zusätzlich das Temperaturprofil eines einzelnen Routers über die gesamte Simulationszeit aufgenommen. Die Ergebnisse sind in Abbildung 3.12 für den Router bei (1,1) (d. h. unten links) in einem 2×2 NoC dargestellt. Prinzipiell spiegelt der Verlauf der Kurven des AMS-basierten Modells auch hier den charakteristischen Kurvenverlauf des SPICE-Modells wider. Abweichungen zwischen den einzelnen Auflösungen für das AMS-Modell sind dabei erneut besonders während der Phasen steigender Temperaturen sichtbar. Auch die eingangs erwähnte Zeitverschiebung von etwa 20 ms ist erkennbar. Eine Erhöhung der Modellierungsgenauigkeit von R0 auf R1 reduziert die durchschnittlichen sowie die maximalen Abweichungen relativ zum SPICE-Modell auf weniger als die Hälfte. Auch die Zahl der Temperaturwerte mit einer Abweichung von weniger als 4°C kann durch eine höhere Auflösung um 6 % vergrößert werden.

Wesentlich effektiver als eine Erhöhung der Auflösung scheint jedoch eine Korrektur der berechneten Temperaturwerte zu sein. In diesem Zusammenhang wurde für die

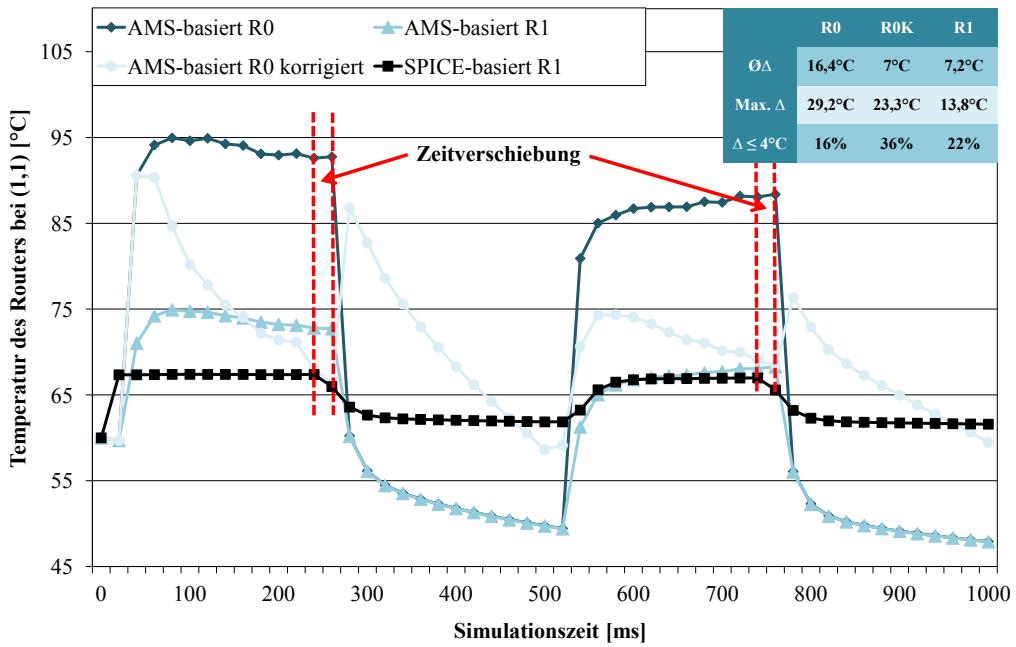


Abbildung 3.12: Temperatur eines einzelnen Routers unter Verwendung verschiedener Auflösungen in einem 2×2 NoC

Auflösung R0 nachträglich ein korrigierter Verlauf R0K berechnet. Die Korrektur erfolgt dabei derart, dass aus den Beträgen der vorangegangenen Abweichungen zwischen dem SPICE-basierten Modell mit der Auflösung R1 und dem AMS-basierten Modell mit der Auflösung R0 ein einfacher gleitender Mittelwert gebildet wird. Die Fenstergröße des Mittelwertes wurde exemplarisch auf 10 festgelegt. Je nachdem, ob der aktuelle Temperaturwert für das AMS-Modell mit R0 im Vergleich zum SPICE-Modell zu hoch oder zu niedrig ausfällt, wird dieser Mittelwert subtrahiert oder addiert.

Wie in Abbildung 3.12 erkennbar ist, reduziert dies die Abweichungen teils deutlich, sodass mit 7°C die durchschnittliche Abweichung unter der des Modells mit der Auflösung R1 liegt. Auch die Anzahl der Abweichungen unterhalb der Grenze von 4°C kann auf diese Weise deutlich gesteigert werden. Problematisch ist jedoch die Korrektur, wenn sich der Verlauf der unkorrigierten AMS-Kurve plötzlich signifikant ändert und sich der SPICE-Kurve annähert. In Abbildung 3.12 ist dies beispielsweise bei 280 ms der Fall. In diesem Fall verschlechtert die Korrektur mithilfe des einfachen gleitenden Mittelwertes das Ergebnis, da der Mittelwert deutlich zu groß ist und sich dem neuen Verhältnis zwischen AMS-Kurve und SPICE-Kurve anpassen muss. Eine Möglichkeit dieses Verhaltens zu kompensieren stellt die Gewichtung des gleitenden Mittelwertes dar. Aktuellere Abweichungen könnten somit stärker in die Berechnung des Korrekturwertes einfließen, um angemessen auf starke Schwankungen des Kurvenverlaufs reagieren zu können.

Tabelle 3.5: Abweichung der durchschnittlichen Temperatur des Systems vom SPICE-Modell unter Verwendung der Auflösung R0 für verschiedene Simulationsperioden T_S

	1 ms	5 ms	10 ms	20 ms
$\bar{\Delta} [^\circ\text{C}]$	9,6	9,6	9,5	9,6
Max. $\Delta [^\circ\text{C}]$	13,1	13,1	13,2	13,2
$\Delta \leq 4^\circ\text{C} [\%]$	0	0	0	0

T_S und T_A Weitere unter dem Aspekt des Zeitaufwands bereits in Abschnitt 3.2.1 betrachtete Faktoren sind die Simulationsperiode T_S und die Abtastperiode T_A . Theoretisch sollte T_S keinen beziehungsweise nur unwesentlichen Einfluss auf die Korrektheit der Modellierung haben, da durch diese Periode lediglich die Frequenz der Strom einspeisung und des Abrufs der Ergebnisse der Temperaturberechnungen variiert werden. Somit beeinflusst T_S lediglich die Aktualität der eingespeisten Ströme und die Anzahl der außerhalb des Modells verfügbaren Berechnungsergebnisse, nicht aber die Gesamtsumme der Ströme beziehungsweise die Genauigkeit der Ergebnisse.

Dass dies tatsächlich der Fall ist, wird in Tabelle 3.5 ersichtlich. In dieser sind die Abweichungen des AMS-Modells von der SPICE-Referenz unter Verwendung verschiedener Werte für T_S dargestellt. Die Abweichungen beziehen sich auf die durchschnittliche Systemtemperatur des zugrunde liegenden NoCs, welches mithilfe der Auflösung R0 modelliert wurde. Im Gegensatz zu den vorherigen Betrachtungen beziehen sich die Ergebnisse auf ein 4×4 NoC, da Systeme dieser Größenordnung als realitätsnäher zu betrachten sind. Zudem ermöglicht nur die gewählte Auflösung R0 eine zügige Simulation des Temperaturverhaltens, während sich feinere Auflösungen besonders für große Systeme in unverhältnismäßig langen Simulationsphasen niederschlagen (siehe Tabelle 3.4 und Abbildung 3.8). Die in Tabelle 3.5 angegebenen durchschnittlichen und maximalen Abweichungen sowie der prozentuale Anteil der unter 4°C liegenden Abweichungen bestätigen die eingangs dargelegte Vermutung der Unabhängigkeit der Korrektheit des Modells von T_S . Demnach ist weder eine positive noch eine negative Beeinflussung möglich. Um allerdings zu klären, ob der zeitliche Verlauf sich in seiner Charakteristik mit dem der SPICE-basierten Simulation deckt und ob die Zeitverschiebung durch unterschiedliche Werte für T_S beeinflusst werden kann, sind weitere Betrachtungen nötig.

In Abbildung 3.13 ist zu diesem Zweck der Temperaturverlauf für einen einzelnen Router unter Verwendung verschiedener Werte für T_S illustriert. Konkret handelt es sich dabei um den zum IPC gehörigen Router, welcher im 4×4 NoC bei (2,2) gelegen ist. Wie zu erkennen ist, folgen auch für verschiedene Werte von T_S alle Kurven grundsätzlich der Charakteristik des SPICE-Modells. Dabei verschiebt sich der Verlauf der Kurve für

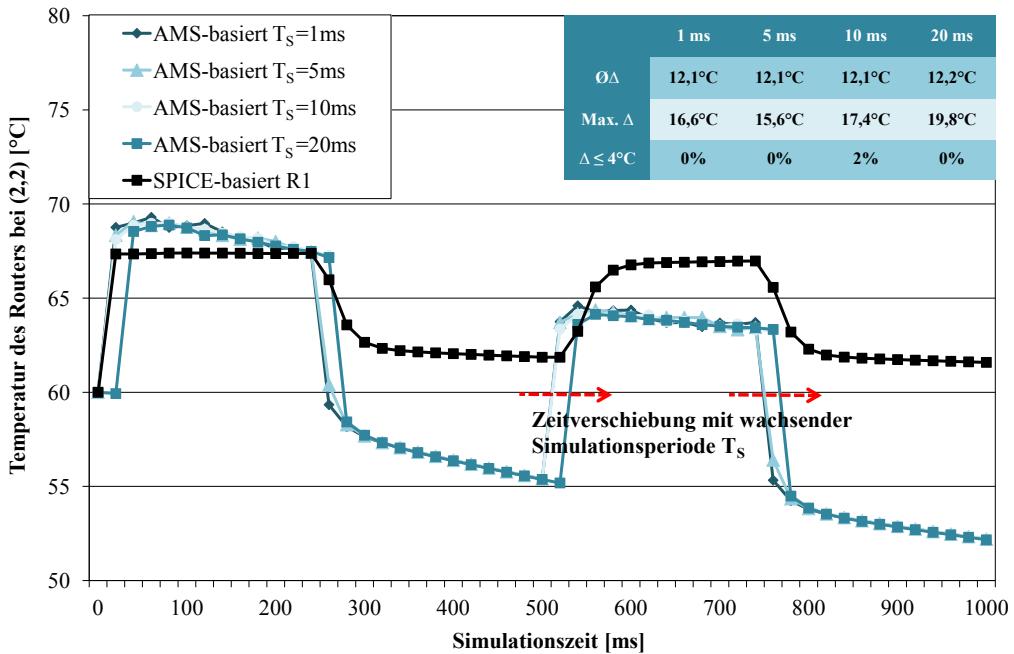


Abbildung 3.13: Ermittelte Temperatur eines einzelnen Routers unter Verwendung verschiedener Simulationsperioden T_S in einem 4×4 NoC

kürzere Simulationsperioden im Vergleich zur Referenzkurve nach vorne, während er sich für längere Perioden zumeist eher an der Referenz orientiert oder sich verzögert. Die Ursache hierfür liegt darin, dass ein hoher Wert für T_S das Anliegen eines aufgrund aktueller Schaltaktivitäten möglicherweise nicht mehr zulässigen Stromes zur Folge hat. Ein niedriger Wert hingegen greift scheinbar der SPICE-Simulation voraus, da diese lediglich auf zu festen Zeitpunkten generierte Aktivitätsstatistiken für die gesamte Simulationszeit zurückgreifen kann. Die durchschnittlichen Abweichungen werden davon nicht beeinflusst. Jedoch nehmen mit wachsender Simulationsperiode die maximalen Abweichungen deutlich zu.

Die Analysen bezüglich des Einflusses von T_A zeigen, dass verschiedene Werte weder die durchschnittliche Temperatur noch die Temperatur einzelner Komponenten des modellierten Systems beeinflussen. Im Zuge der Untersuchungen wurden dabei Abtastperioden von $50 \mu\text{s}$, $100 \mu\text{s}$, $500 \mu\text{s}$ und 1 ms betrachtet. Dieses Verhalten ist ein Indiz dafür, dass die gewählten Werte einerseits zu klein sein könnten, um signifikante Verschlechterungen der Genauigkeit zu bewirken. Gegen eine weitere Erhöhung sprechen allerdings der bereits bei einem Wert von $500 \mu\text{s}$ stagnierende Zeitvorteil (siehe Abbildung 3.9) sowie die Beschränkung durch T_S . Andererseits vergrößert eine weitere Verringerung der Abtastperiode auf wenige Millisekunden oder gar Nanosekunden den zeitlichen Aufwand. Der Trend in Abbildung 3.9 lässt daher vermuten, dass eine möglicherweise erhöhte Genauigkeit in keiner Relation zum geleisteten Mehraufwand steht.

3.2.3 Qualitative Bewertung und Zusammenfassung

Das in diesem Kapitel vorgestellte und in den Abschnitten 3.2.1 und 3.2.2 bezüglich seiner Leistungsfähigkeit und Modellierungsgenauigkeit bewertete Temperaturmodell für NoC-basierte Systeme beruht auf dem Dualismus zwischen dem Fluss elektrischer und thermischer Energie. Ein Vorteil des mithilfe der SystemC-eigenen AMS-Bibliothek implementierten Modells gegenüber SPICE-basierten Simulationen liegt im zeitlichen Ablauf der Berechnung des Temperaturverlaufs und den daraus resultierenden Einsatzmöglichkeiten für ein temperaturorientiertes Systemmanagement. Die SPICE-Modelle greifen für die Berechnung des Temperaturverlaufs auf zuvor in funktionalen Simulationen ermittelte Aktivitätsprofile der einzelnen Komponenten des zu modellierenden Systems zurück. Eine Modellierung der Temperaturentwicklung des Chips für einen bestimmten Zeitraum kann erst stattfinden, nachdem die funktionale Simulation über diesen Zeitraum vollständig absolviert wurde. Die resultierende sequentielle Abfolge von Aktivitätsermittlung und anschließender Temperaturberechnung disqualifiziert derartige Modelle für ein Temperaturmanagement zur Laufzeit.

Das AMS-basierte Modell ermöglicht hingegen einen parallelen Ablauf von funktionaler Simulation und Temperaturberechnung. Somit können parallel zur funktionalen Simulation die thermischen Auswirkungen von Schaltaktivitäten berechnet werden, um während der Simulation des laufenden Betriebs des zugrunde liegenden Systems entsprechende Maßnahmen zu treffen. Zudem ist das AMS-Modell aufgrund der Integration in eine Simulationsumgebung unabhängig von externen Werkzeugen. Diese müssen im Rahmen SPICE-basierter Modelle oft eingesetzt werden, um für die Temperaturberechnung benötigte Statistiken zu erfassen.

Unter Ausnutzung der Trägheit von Temperaturanstieg und -ausbreitung (siehe Abschnitt 2.4.1) können bei ausreichender Leistungsfähigkeit des Modells Temperaturänderungen unter Umständen berechnet werden, bevor sie real in Erscheinung treten. Auf diese Weise können proaktive Maßnahmen eingeleitet werden, um unerwünschten Temperaturwerten bereits im Vorfeld entgegenzuwirken. Um ein solch leistungsfähiges Modell gewährleisten zu können, sind neben der zeitlichen Organisation weitere Aspekte wie Modellierungsgenauigkeit und die Wahl von T_S und T_A von Bedeutung. Eine kurze qualitative Bewertung des Einflusses dieser Faktoren ist in Tabelle 3.6 zusammengefasst.

Die verschiedenen Modellierungsauflösungen beeinflussen besonders den Zeitaufwand, welcher für das Setup des Temperaturmodells (d. h. die Elaboration) nötig ist. Lediglich die größte Auflösung R0 liefert insbesondere im Hinblick auf größere zu modellierende Systeme (z. B. 6×6 NoC) akzeptable Ergebnisse. Dies gilt auch für die eigentliche Simulationsphase, bei welcher die Dauer der Temperatursimulation für NoC-Größen bis 6×6 nur bei Verwendung der Auflösung R0 deutlich unterhalb der simulierten Systemzeit liegt. Der zeitliche Mehraufwand höherer Auflösungen bewegt sich zwar eher

im Sekundenbereich. Dies verhindert jedoch bereits ein mögliches proaktives Management, da die Dauer der Simulation damit die simulierte Systemzeit überschreitet. Die Korrektheit des Kurvenverlaufs, genauer gesagt die Charakteristik des Temperaturanstiegs in Phasen hoher Aktivität und der Abkühlung in Leerlaufphasen, ist im Vergleich zur SPICE-Referenz für alle Auflösungen gegeben. Jedoch ist die absolute Genauigkeit des R0-Modells bezogen auf das SPICE-Modell als mangelhaft zu bewerten. Höhere Auflösungen erzielen eindeutig bessere Ergebnisse, wobei auch für diese Abweichungen bestehen bleiben. Zwischen R1 und R2 sind keine nennenswerten Unterschiede feststellbar. Von primärem Interesse ist allerdings ohnehin der charakteristische Verlauf, da nur auf dessen Basis Temperaturtrends abgeschätzt werden können. Absolute Modellgenauigkeiten können durch entsprechende Skalierungen oder Korrekturwerte kompensiert werden. Nicht zu vernachlässigen ist in diesem Fall auch die in Abbildung 3.7 illustrierte, abweichende Realisierung der Stromquellen unter SystemC AMS und SPICE. Diese trägt ebenfalls zu unterschiedlichen Temperaturprofilen bei.

Unabhängig von der gewählten Modellierungsauflösung weist das AMS-Modell im Vergleich zur Referenzkurve eine zeitliche Verschiebung auf. Diese scheint aufgrund der konstanten Differenz systematischer Natur zu sein und ist vermutlich ebenfalls auf die unterschiedliche Implementierung der Stromquellen zurückzuführen. Diese Verschiebung lässt sich mithilfe der Simulationsperiode T_S beeinflussen, da diese die Häufigkeit bestimmt, mit der Stromwerte in das RC-Netz eingespeist werden. Kurze Werte für T_S verringern dabei die Abweichung des AMS-Modells von der Referenz, da die Zeitspanne verkürzt wird, für welche aus zwei diskreten Werten eine kontinuierliche Funktion gebildet wird. Allerdings führt dies zu einem unverhältnismäßig hohen Zeitaufwand für die Simulation (siehe Diagramm 3.9). Hohe Werte für T_S reduzieren hingegen den Zeitaufwand für die Simulation drastisch. Diese führen jedoch zu jener Zeitverschiebung und dazu, dass am Anfang einer Periode aufgetretene Schaltaktivitäten erst sehr spät in die Temperaturberechnung einfließen können. Somit muss ein Kompromiss zwischen zeitlichem Aufwand und der Aktualität der Temperaturwerte getroffen werden. Die Genauigkeit des Modells oder den Zeitaufwand für die Elaboration beeinflusst T_S hingegen nicht. Dies gilt auch für die Abtastperiode T_A des Temperaturmodells. Lange Perioden sorgen für einen geringeren Zeitaufwand, da die internen Funktionsaufrufe des Modells seltener stattfinden. Theoretisch sollte dies auch zu Genauigkeitsabweichungen führen. Die für die Untersuchungen vermutlich relativ klein gewählten Perioden haben jedoch keinen messbaren Einfluss auf die Genauigkeit. Da eine Erhöhung von T_A die Modellgenauigkeit voraussichtlich verringert und für die größeren der geprüften Werte bezüglich der Zeitersparnis bereits eine Sättigung eintritt, wird auf weitere Untersuchungen verzichtet. Hohe Werte für T_A sind demnach die beste Wahl, da sich die Leistungsfähigkeit ohne nennenswerte Nachteile steigern lässt. Jedoch ist darauf zu achten, dass T_A den

Tabelle 3.6: Qualitative Zusammenfassung der in den Abschnitten 3.2.1 und 3.2.2 gewonnenen Erkenntnisse bezüglich der Modellierungsauflösungen R0, R1 und R2 und der Parameter T_S und T_A (Legende: + gut, \circ neutral, - schlecht, \times kein Einfluss)

	R0	R1	R2	T _S		T _A	
				lang	kurz	lang	kurz
Zeitaufwand Elaborationsphase	+	-	-	\times	\times	\times	\times
Zeitaufwand Simulationsphase	+	\circ	-	+	-	+	-
Korrektheit des Kurvenverlaufs	+	+	+	\times	\times	\times	\times
Absolute Genauigkeit	-	\circ	\circ	\times	\times	\times	\times
Zeitliche Verschiebung	\times	\times	\times	\circ	+	\times	\times

Wert von T_S nicht überschreitet. Andernfalls werden die Temperaturprofile seltener berechnet als neue Aktivitätsstatistiken bereitgestellt vorliegen.

Die Voraussetzung für die Nutzbarkeit eines Temperaturmodells für den Einsatz in einem proaktiven temperaturorientierten Systemmanagement ist eine schnelle und dennoch hinreichend korrekte Modellierung des charakteristischen Kurvenverlaufs von Temperaturanstiegen und Abkühlungen. Die durchgeführten Untersuchungen haben gezeigt, dass sich das in diesem Kapitel vorgestellte SystemC AMS-basierte Temperaturmodell diesbezüglich als geeignet erweist. Bezuglich der absoluten Genauigkeit müssen allerdings Abstriche gemacht werden. Allerdings konnte ebenso gezeigt werden, dass korrigierende Maßnahmen (z.B. gleitender Durchschnitt vorangegangener Abweichungen) die Abweichungen auf einfache Weise deutlich reduzieren können. Das Systemmanagement soll das Modell nutzen, um auf Basis von zur Systemlaufzeit detektierten Schaltaktivitäten die Temperaturrentwicklung des Systems möglichst weit im Voraus zu berechnen. Die angestellten Berechnungen sollen wiederum für proaktive Managemententscheidungen genutzt werden. Die wichtigste Eigenschaft des Temperaturmodells ist dessen Fähigkeit parallel zu einer funktionalen Simulation, später auch parallel zu einem realen System, den Temperaturverlauf desselben berechnen zu können.

Während des Vergleichs mit einem identischen SPICE-basierten Modell verdeutlichten sich allerdings einige die Korrektheit des Modells beeinträchtigende Abweichungen. Einerseits ist es möglich diese größtenteils implementierungsbedingten Abweichun-

gen durch die Anpassung einzelner Modellparameter zu minimieren. Andererseits ist das Modell nichtsdestotrotz in der Lage den charakteristischen Verlauf der Temperaturkurven des Referenzmodells effizient und schnell nachzubilden. Zukünftige Temperaturtrends können somit ausreichend zuverlässig vorausberechnet werden. Das AMS-basierte Modell bietet zwar eine niedrigere Genauigkeit als die SPICE-Realisierung, erfordert dafür allerdings auch einen geringeren zeitlichen Aufwand. Damit bildet das Temperaturmodell einen wichtigen Grundstein für das proaktive Management NoC-basierter Mehrkernprozessoren. So müssen beispielsweise kostenintensive Temperatursensoren nur noch gelegentlich für Rekalibrierungen beansprucht werden oder könnten teilweise sogar eliminiert werden. Die auf Grundlage der Daten dieser Sensoren durchgeführten, vergleichsweise langsamen reaktiven Gegenmaßnahmen im Fall unerwünschter Temperaturwerte würden somit durch proaktive Maßnahmen mit verkürzter Antwortzeit ersetzt.

Der Aufwand für die Implementierung des Temperaturmodells beläuft sich auf insgesamt etwa 3500 Zeilen Code. Davon entfallen allein 2200 Zeilen auf das eigentliche Temperaturmodell, die restlichen 1300 Zeilen realisieren die Simulationsumgebung sowie die Integration des Temperaturmodells. Berücksichtigt man zusätzlich den funktionalen Teil der Simulation kommen weitere 2800 Zeilen Code hinzu. Diese Zahlen verdeutlichen nochmals den für Temperaturmodell aufgebrachten Implementierungsaufwand.

4 Temperaturorientiertes Management NoC-basierter Systeme

Dieses Kapitel beschäftigt sich mit dem Management NoC-basierter Systeme. Den Ausgangspunkt bilden die in Kapitel 2 aufgeführten Sachverhalte und daraus resultierende Problemstellungen. Der Fokus liegt auf einem temperaturorientierten Konzept, da eine entsprechende Temperatursteuerung durch die Minderung und Verzögerung von Alterungs- und Verschleißerscheinungen (siehe Abschnitte 2.1.2 und 2.2.3) besonders zu einer Steigerung der Zuverlässigkeit beiträgt. Im Rahmen dieser Arbeit werden diesbezüglich Konzepte für ein dynamisches Temperaturmanagement sowie ein dynamisches Task-Mapping zur Laufzeit NoC-basierter Systeme untersucht. Für Ersteres ist die Anpassung leistungsrelevanter Systemparameter zur positiven Beeinflussung des Temperaturprofils vorgesehen. Für Letzteres soll ein temperaturreduzierendes Arrangement der auf den Systemkomponenten bearbeiteten Tasks erfolgen. Die Besonderheit dieser Konzepte ist dabei ihre proaktive Auslegung. Sämtliche Entscheidungen stützen sich auf eine Prognose der zukünftigen Entwicklung des Temperaturprofils. Auf diese Weise können Gegenmaßnahmen bereits vor dem Auftreten unerwünschter Temperaturen eingeleitet werden. Die Grundlage dafür ist das in Kapitel 3 vorgestellte Temperaturmodell. Dieses berechnet auf Basis registrierter Schaltaktivitäten das Temperaturprofil voraus und macht sich die Trägheit der Wärmeausbreitung zunutze, um Vorhersagen zu treffen.

Die eigentlichen Steuerungsmaßnahmen sind an herkömmliche Management- und Mapping-Methoden angelehnt. Ein Überblick über beeinflussbare Systemparameter, mögliche Methoden zur Beeinflussung dieser sowie existierende Managementsysteme und -konzepte wird in Abschnitt 4.1 gegeben. Im Anschluss werden die proaktiven Konzepte für ein temperaturorientiertes Systemmanagement und Task-Mapping in den Abschnitten 4.2 und 4.3 vorgestellt und mit herkömmlichen Methoden verglichen. Eine Zusammenfassung der erzielten Erkenntnisse rundet den jeweiligen Abschnitt ab.

4.1 Stand der Technik

Im Zuge eines temperaturorientierten Systemmanagements ist eine Reduzierung der Leistungsfähigkeit unumgänglich, da nur auf diese Weise die Leistungsaufnahme und damit die erzeugte Wärme reduziert werden können. Leistungsfähigkeit, Leistungsaufnah-

me und Temperatur sind sowohl untereinander als auch mit zuverlässigkeitbezogenen Aspekten eng verknüpft. Daher müssen Kompromisse eingegangen werden, um alle Faktoren angemessen zu berücksichtigen. Aus diesem Grund ist eine differenzierte Sicht auf die beeinflussbaren Parameter, die verwendeten Methoden und die resultierenden Konsequenzen notwendig.

Im Folgenden wird eine Klassifizierung vorgestellt, welche sich auf die wichtigsten Ansätze und Methoden beschränkt. Besonderer Wert wird dabei auf zur Laufzeit anwendbare dynamische Methoden gelegt. Unter Berücksichtigung des verteilten und heterogenen Aufbaus NoC-basierter Systeme ergeben sich zunächst die Möglichkeiten eines regulierenden Eingriffs auf Systemebene oder der gezielten Steuerung einzelner Systemkomponenten. Grundsätzlich bieten sich für die Beeinflussung der Temperaturverteilung eine Beschränkung der Leistungsfähigkeit sowie eine Regulierung der Lastverteilung an, um auf diese Weise die Leistungsaufnahme zu reduzieren.

Steuerung der Schaltungsparameter Die Leistungsaufnahme lässt sich sowohl auf Systemebene für die Gesamtheit aller Komponenten als auch für einzelne Komponenten reduzieren, um auf diese Weise die Wärmeabstrahlung zu verringern. Dabei kann mit geeigneten Methoden auf alle drei Bestandteile der Gesamtleistung, also auf die dynamische, die statische und die durch Kurzschlüsse verursachte Leistungsaufnahme (siehe Gleichung 2.1), Einfluss genommen werden. Eine gezielte Beeinflussung einzelner Komponenten hat dabei den Vorteil, dass bereits vorherrschende Ungleichgewichte berücksichtigt werden können. Auf diese Weise kann eine Beeinflussung solcher Komponenten vermieden werden, bei denen dies gar nicht nötig ist (z. B. Bereiche niedriger Temperatur) oder aufgrund gewisser Vorgaben sogar unerwünscht ist (z. B. Komponenten mit Echtzeitanforderungen). Um die einzelnen Komponenten der Leistungsaufnahme zu reduzieren, eignen sich verschiedene Methoden.

Am weitesten verbreitet sind DVS und DFS zur dynamischen Anpassung der Betriebsspannung V_{DD} und der Taktfrequenz [SSH⁺03]. Durch DVS können alle drei Bestandteile der Gesamtleistung reduziert werden, der dynamische Anteil sogar quadratisch. DFS dagegen reduziert lediglich den dynamischen sowie den durch Kurzschlüsse verursachten Anteil linear. Die Reduzierung von V_{DD} kann allerdings nur in begrenztem Maß genutzt werden, da eine niedrige Versorgungsspannung die Signalintegrität sowie die Flankensteilheit negativ beeinflusst (siehe Abschnitt 2.1.2). Letzteres hat durch verzögerte Umladevorgänge einen negativen Einfluss auf die Leistungsfähigkeit einer Schaltung. DFS beeinflusst hingegen lediglich die Geschwindigkeit einer Schaltung und somit die Dauer t_{op} einer auszuführenden Operation. Um leichter eine Aussage über Sinnhaftigkeit und Effektivität einer Methode treffen zu können, ist das **Power-Delay-Product** ($PDP = P_{Gesamt} \cdot t_{op}$), also die für die Ausführung einer Operation aufgebrachte

Energie, besser geeignet als eine alleinige Betrachtung der Leistungsaufnahme, da hierbei auch die Ausführungszeit berücksichtigt wird. Für DVS ist durch den quadratischen Einfluss auf die dynamische Leistung und einen lediglich linearen negativen Einfluss auf die Transistorschaltzeiten [RCN03] eine Verbesserung des PDP mit reduzierter Versorgungsspannung zu erwarten. Dies gilt allerdings nur solange V_{DD} einen gewissen Wert nicht unterschreitet und das Verhältnis zwischen V_{DD} und der Schwellspannung gewahrt bleibt. Andernfalls reduziert sich ungewollt die Taktfrequenz. Die bloße Reduzierung der Taktfrequenz durch DFS wiederum beeinflusst die Leistungsaufnahme und die Ausführungszeit lediglich linear. Somit ist zwar keine Reduzierung der Energie und damit des PDP zu erwarten. Jedoch lässt sich die Temperaturentwicklung durch DFS vorhersagbar und abgesehen von einer reduzierten Leistungsfähigkeit ohne unerwünschte Nebeneffekte steuern [SSH⁺03].

Die Technik des Local Toggling [SSH⁺03] bietet eine weitere Möglichkeit die Temperatur zu beeinflussen. Dabei werden die einzelnen Systemkomponenten normal betrieben, solange sie keinen unzulässigen Temperaturen oder thermischem Stress ausgesetzt sind. Tritt dieser Fall dennoch ein, werden die betreffenden Komponenten zur Abkühlung in einen vorübergehenden Ruhezustand versetzt.

Auf ähnliche Weise reduzieren Clock und Power Gating die Leistungsaufnahme einzelner Komponenten und wirken auf diese Weise positiv auf das Temperaturprofil des Gesamtsystems ein. Clock Gating trennt dabei alle sich im Leerlauf befindlichen Systemkomponenten vom Taktnetz. Nach Gleichung 2.1 reduziert sich damit die Leistungsaufnahme nicht getakteter Komponenten auf die durch Leckströme verursachte statische Leistung P_{Leck} . Die Granularität des Clock Gatings, mit welcher beispielsweise ganze NoC-Router oder nur einzelne Komponenten abgeschaltet werden, hat maßgeblichen Einfluss auf die dynamische Leistungsaufnahme. Untersuchungen zeigen, dass feiner aufgelöste Ansätze diesbezüglich entscheidende Vorteile bringen [C.11]. Power Gating hingegen trennt untätige Komponenten nicht nur vom Taktnetz sondern auch von der Stromversorgung, um die gesamte Leistungsaufnahme auf ein Minimum zu reduzieren. Allerdings sorgen die zusätzlich benötigte Logik für die Abschaltung und für die Wiederherstellung des Zustandes vor der Abschaltung sowie eine Verzögerung bei der Rückkehr in den aktiven Betrieb für eine begrenzte Einsetzbarkeit [AE03]. Dies gilt insbesondere für NoC-basierte Systeme, bei denen aufgrund der Anzahl der Systemkomponenten und deren möglicher Heterogenität die Identifikation von Leerlaufphasen problematisch sein kann.

Eine weitere Methode der Temperaturbeeinflussung zur Laufzeit ist das Body Biasing. Dieses verändert mithilfe der Modifikation des Potentials der Substratschicht eines Transistors dessen Schwellspannung [MFMB02] und beeinflusst damit das Schaltverhalten. Eine erhöhte Schwellspannung führt dabei einerseits zu reduzierten Leckströmen.

Andererseits verringert sich aber auch die Leistungsfähigkeit des Transistors. Die Bereitstellung eines variablen Potentials der Substratschicht erhöht zudem den Flächenbedarf eines Transistors.

Im Rahmen dieser Arbeit wird aufgrund der einfachen Implementierbarkeit lediglich DFS für den Nachweis der Funktionsfähigkeit der vorgestellten Konzepte eingesetzt. Neben dem niedrigen Realisierungsaufwand spricht auch der Einsatz solcher Techniken in kommerziellen Produkten wie Intel SpeedStep [Int04] oder AMD Cool 'n' Quiet für diese Wahl.

Regulierung der Lastverteilung Im Zuge einer Regulierung der Lastverteilung ist hingegen eine Betrachtung auf Systemebene notwendig, da eine effektive Verlagerung von Aufgaben zwischen den einzelnen Systemkomponenten unter anderem die Kenntnis über die derzeitige Lastsituation voraussetzt. Bei dieser Art des Systemmanagements kommt es nicht zwangsläufig zu einer Beeinträchtigung der Leistungsfähigkeit. Stattdessen wird eine lokale Reduzierung der Leistungsaufnahme durch eine Umverteilung der zu bearbeitenden Aufgaben erreicht. Dabei kann unter Umständen auch das PDP durch eine parallele Abarbeitung der Aufgaben positiv beeinflusst werden. Voraussetzung dafür ist, dass die erzielte Zeitersparnis nicht durch eine erhöhte Leistungsaufnahme kompensiert wird.

Allerdings gestaltet sich ein solches Management unter Umständen recht kompliziert, da die mithilfe des Systems bearbeiteten Aufgaben sehr komplex und vielfältig sein können und während der Entwurfsphase nicht zwingend bekannt sind. Zudem können die Aufgaben aus einer variierenden Anzahl untereinander abhängiger Teilaufgaben, sogenannter Tasks, bestehen. Daher werden verschiedene Strategien genutzt, um die Tasks bereits während der Entwurfsphase des Systems möglichst optimal auf den geeigneten Systemkomponenten zu verteilen. Dies geschieht nach bestimmten Gesichtspunkten wie beispielsweise der Maximierung des Datendurchsatzes [LWH⁺05] oder einer Reduzierung von Temperaturspitzen und -ungleichgewichten [HAQT⁺04, CRWG08]. Jedoch existieren auch Ansätze, welche die Platzierung der Tasks dynamisch zur Laufzeit des Systems vornehmen [HHKS08, SJKS10]. Der Vorteil dabei ist, dass auch der Zustand des Systems sowie die Verfügbarkeit einzelner Systemkomponenten in die Mapping-Entscheidung einbezogen werden können.

Für die statische Verteilung der Tasks während der Entwurfsphase und die dynamische Umverteilung der Tasks zur Laufzeit des Systems kommen auf verschiedene Aspekte optimierte Algorithmen zum Einsatz. Diese berücksichtigen beispielsweise eine möglichst gleichmäßige Auslastung aller Komponenten [CRWG08], eine Minimierung der Übertragungswege untereinander kommunizierender Komponenten (Nearest-Neighbor-Prinzip) [KSS11] oder ein Ausbalancieren des Temperaturprofils bei der Ver-

teilung der Aufgaben [KBB06]. Diese Thematik wird eingehender in Abschnitt 4.3 behandelt.

Existierende Konzepte Die genannten Punkte und eine Reihe weiterer Methoden lassen sich zu Ansätzen für ein Systemmanagement zusammenfassen. Das Verständnis eines solchen Managements als Teil eines herkömmlichen Betriebssystems ist allerdings für kommunikationsorientierte, dezentral ausgelegte Architekturen schwer anwendbar und nicht mehr zeitgemäß. Ein auf NoC-basierte Systeme ausgelegtes Managementkonzept muss neben der Regulierung der eingangs aufgeführten Parameter und den damit einhergehenden Effekten und Wechselwirkungen eine Vielzahl weiterer Faktoren berücksichtigen. Hierzu zählen neben dem verteilten Charakter der Architektur und der höheren Komplexität auch eventuelle Anforderungen an die QoS oder Echtzeitfähigkeit des zugrunde liegenden Systems. Erschwerend kommen sich dynamisch ändernde Rahmenbedingungen hinzu. Software-seitig betrifft dies vor allem wechselnde Applikationen und somit variierende Lastprofile. Auf Seiten der Hardware sind nicht oder nur begrenzt verfügbare Komponenten (z. B. durch Ausfall, temporäre Abschaltung oder Derating) betroffen. Somit muss ein entsprechendes temperaturorientiertes Management zur Laufzeit sowohl die abstrakte Funktionalität des Systems als auch den Zustand der zugrunde liegenden Hardware berücksichtigen, um die Temperatur effektiv zu beeinflussen und gleichzeitig die Leistungsfähigkeit nicht zu stark zu beeinträchtigen. Es existiert bereits eine Vielzahl von Konzepten für ein auf NoCs zugeschnittenes laufzeitfähiges Systemmanagement. Einige nennenswerte Vertreter sind nachfolgend aufgeführt.

Die Konzepte in [CBR⁺05, CGRB06, GNR⁺10] konzentrieren sich auf die Überwachung der Parameter NoC-basierter Systeme, um Informationen für ein Management zur Verfügung zu stellen. In [CBR⁺05] wird dazu ein Konzept für eine ereignisbasierte Überwachung von NoC-Komponenten mithilfe physikalisch vorhandener Sonden vorgestellt. In [CGRB06] werden Möglichkeiten für die Integration dieser in ein NoC betrachtet.

In [GNR⁺10] wird ein hierarchisch organisiertes Netz von Überwachungseinheiten für SoCs vorgeschlagen, um alle Architekturniveaus und Abstraktionsstufen in die Überwachung verschiedener Parameter einzubeziehen. Auf diese Weise soll die Leistungs- und Einsatzfähigkeit des Gesamtsystems auch unter wechselnden Bedingungen oder im Fehlerfall gewährleistet werden. Für die Verwendung der Informationen für eine Steuerung werden in dieser Arbeit keine Vorschläge geliefert.

In [RIT07] wird auf ähnliche Weise ein hierarchisches, mehrstufiges Netz von Überwachungsinstanzen eingeführt und für ein reaktives Systemmanagement NoC-basierter Systeme eingesetzt. Gegenstand der Überwachung sind für die auf den Systemkomponenten ausgeführten Applikationen relevante Parameter. Der Zweck dieses Managements ist die Erhöhung der Fehlertoleranz der Applikationen durch eine Verlagerung

dieser heraus aus defekten in funktionsfähige Bereiche des NoCs. Eine entsprechende Neukoordination der Datenströme wird ebenfalls betrachtet.

Auch in [ERHH11] wird ein hierarchisches Konzept vorgestellt. Dieses wird für ein temperaturorientiertes Management von Systemen genutzt, welche auf 3D-NoCs basieren. Im Detail wird eine Task-Migration aus heißen in kalte Chip-Bereiche durchgeführt, um Temperaturspitzen zu minimieren. Die Temperaturen werden durch nicht näher erläuterte Sensoren gemessen.

Ein Ansatz für eine reaktive Überwachung und Steuerung der Temperatur NoC-basierter Systeme wird in [AM09] vorgestellt. Nicht näher spezifizierte Sensoren überwachen die Temperatur der einzelnen Systemkomponenten und nutzen das NoC, um etwaige Grenzwertüberschreitungen einer zentralen Steuereinheit mitzuteilen. Diese reagiert mit dem Versenden entsprechender Instruktionen für DFS und DVS.

In [SPKJ04] wird ein Konzept für ein Temperaturmanagement für SoCs vorgestellt. Dieses greift auf ein Temperaturmodell zurück, welches dem in dieser Arbeit entwickelten Modell ähnelt, und setzt dieses für reaktive und proaktive Steuerungsmaßnahmen ein. Die proaktiven Maßnahmen beruhen dabei auf einer Abschätzung der typischen Leistungsaufnahme der betreffenden Systemkomponenten. Die Abschätzungen werden anhand der zugrunde liegenden Netzwerkarchitektur, der verwendeten Technologie und der während einer funktionalen Simulation des Systems aufgezeichneten Netzwerkaktivitäten getroffen. Als Steuerungsmaßnahmen kommen reaktives und proaktives Routing zum Einsatz.

[CRG08] nutzt hingegen die Methode autoregressiver gleitender Durchschnitte, um die Temperaturentwicklung von SoCs auf Basis vorheriger Temperaturmessungen durch thermische Sensoren vorauszusagen. Diese Informationen werden für eine Zuweisung und Umverteilung von Tasks auf die Systemkomponenten genutzt, um auf diese Weise das Temperaturprofil des Systems auszubalancieren. Für dieses Konzept werden dauerhaft Temperatursensoren für die Messung und die Korrektur benötigt.

4.2 Anwendungsszenario I: Proaktives Temperaturmanagement

In diesem Abschnitt wird ein proaktiver Ansatz für ein temperaturorientiertes Systemmanagement zur Laufzeit NoC-basierter Mehrkernprozessoren vorgestellt. Dieser Ansatz nutzt das in Kapitel 3 gezeigte Temperaturmodell und macht sich die Trägheit eines Temperaturanstiegs nach dem Auftreten einer Schaltaktivität zunutze, um die Auswirkung der Aktivität auf das Temperaturprofil des Systems im Vorfeld abzuschätzen. In Abschnitt 4.2.1 werden die Realisierung des proaktiven Managementsystems sowie die Eingliederung in die Simulationsumgebung betrachtet, welches bereits um das nö-

tige Temperaturmodell erweitert wurde. Zusätzlich wird ein zu Vergleichszwecken implementiertes reaktives Managementsystem eingeführt. Im Anschluss befasst sich Abschnitt 4.2.2 mit dem Vergleich zwischen proaktivem und reaktivem Systemmanagement und konzentriert sich dabei auf die Auswirkungen des proaktiven Ansatzes auf das Temperaturprofil sowie die Leistungsfähigkeit des zugrunde liegenden Systems. Eine Zusammenfassung findet in Abschnitt 4.2.3 statt.

4.2.1 Konzept

In diesem Abschnitt wird der Entwurf für das proaktive, temperaturorientierte Systemmanagement vorgestellt. Zunächst werden der prinzipielle Ablauf des Managements sowie eine entsprechende Erweiterung der in Abschnitt 3.1.3 angeführten Simulationsumgebung erläutert. Im Anschluss wird die interne Funktionalität derjenigen Komponente eingehender beleuchtet, welche für die eigentliche Steuerung verantwortlich ist. Diese Komponente wird im Folgenden als Thermal Management Unit (TMU) bezeichnet. Aus Gründen der Komplexität beschränkt sich diese Arbeit auf das Konzept einer einzelnen zentralen TMU. Entwürfe, welchen ein zuverlässigeres Cluster-basiertes TMU-Konzept zugrunde liegt, sind aber ebenfalls denkbar. Auf solche verteilten Konzepte wird auch deshalb bewusst verzichtet, da die Fähigkeit des Systems Temperaturtrends korrekt zu prognostizieren und proaktiv entsprechende Maßnahmen einzuleiten im Mittelpunkt der Betrachtungen steht. Weiterhin werden Überlegungen bezüglich der Realisierung und der Integration des, für eine temperaturorientierte Steuerung notwendigen, Überwachungssystems angestellt. Der dargelegte Ablauf der Ausführungen trifft ebenfalls für die reaktive Variante des Systemmanagements zu. Dieses wird als Referenz benötigt, um die Stärken und Schwächen des proaktiven Ansatzes zu identifizieren.

Proaktives Temperaturmanagement

Das vorgestellte Konzept eines proaktiven Systemmanagements basiert auf der Überwachung ausgewählter Systemparameter. Für den Fall, dass die Werte dieser Parameter vorgegebene Bereiche verlassen, werden durch das Management entsprechende Reaktionen eingeleitet. Bezogen auf das dynamische, proaktive Temperaturmanagement NoC-basierter Systeme bedeutet dies, dass die Temperatur aller Systemkomponenten überwacht wird und durch die TMU ausgewertet werden muss. Basierend auf den gewonnenen Informationen können individuelle Instruktionen an die einzelnen Komponenten übertragen werden, um vorgegebene Direktiven umzusetzen.

Prinzipieller Ablauf Der schematische Ablauf des proaktiven Managements und die Integration desselben in die Simulationsumgebung sind in Abbildung 4.1 illustriert. Die

Besonderheit dieses Konzepts ist, dass für eine temperaturorientierte Steuerung des Systems die Temperatur der Komponenten nicht durch Temperatursensoren gemessen wird. Vielmehr kommt das in Kapitel 3 beschriebene Temperaturmodell zum Einsatz. Wie erkennbar ist, verfügt die Simulationsumgebung über zwei separate Temperaturmodelle. Das als „Tatsächliches Profil“ gekennzeichnete Modell verhält sich identisch zu dem in Abschnitt 3.1.3 vorgestellten Modell, repräsentiert den tatsächlichen Verlauf der Temperaturentwicklung des NoC-basierten Systems und dient dem Abgleich mit dem prognostizierten Verlauf. Das Profil wird mithilfe der Aktivitätsstatistiken berechnet, welche durch die Simulationsumgebung während der funktionalen Simulation aufgezeichnet und dem Modell periodisch nach Ablauf der Simulationsperiode T_S zur Verfügung gestellt werden.

Parallel zu diesem Modell existiert ein zweites als „Vorhersageprofil“ bezeichnetes Temperaturmodell, welches den durch die TMU berechneten Temperaturtrend repräsentiert. Die dafür nötigen Aktivitätsprofile werden dem Modell durch die TMU zur Verfügung gestellt. Diese wiederum wird von dem Überwachungssystem mit den entsprechenden Statistiken versorgt, welches in das simulierte NoC integriert ist. Eine Ausgabe des prognostizierten Temperaturverlaufs an die TMU erfolgt periodisch nach Ablauf der TMU-internen Simulationsperiode $T_{S,TMU}$. Die Häufigkeit, mit welcher die TMU aktualisierte Informationen über die Temperaturentwicklung erhält, ist damit relativ frei wählbar. Auf diese Weise kann untersucht werden, inwiefern sich ein zeitlicher Vorsprung für die TMU vor dem Verlauf des tatsächlichen Profils (siehe Abschnitt 3.2.1) erzielen lässt. Dieser wird in jedem Fall durch einen erhöhten Berechnungsaufwand erkauft.

Das angesprochene Überwachungssystem besteht aus einem Netz von Sonden, welche die Aktivität der ihnen zugewiesenen Komponenten beobachten und bei Überschreitung eines definierten Schwellwertes $Akt_{Schwell}$ die Aktivitäten sowie den Beobachtungszeitraum der TMU melden. Genauere Erläuterungen bezüglich der Anzahl und der Platzierung der Sonden sowie deren Anbindung an das On-Chip-Netzwerk erfolgen im späteren Verlauf.

Stellt die TMU anhand der Auswertung des Vorhersageprofils einen unerwünschten Temperaturtrend fest, so werden Steuerungsinstruktionen generiert und an die betroffenen Komponenten versendet. Eine Besonderheit ist, dass die Kommunikation zwischen Überwachungssystem und TMU sowie die Kommunikation zwischen TMU und den Systemkomponenten in die funktionale Simulation integriert wird. Im Detail bedeutet dies, dass diese Kommunikationen im Rahmen der funktionalen Simulation als reale Datenübertragungen modelliert werden. Auf diese Weise kann der Einfluss des managementbezogenen Datenverkehrs auf die Lastsituation im NoC und auf die generierte Wärme direkt in der Simulation berücksichtigt werden. Dies erlaubt eine Bewertung des proaktiven Managements bezüglich seiner Effektivität und Praxistauglichkeit.

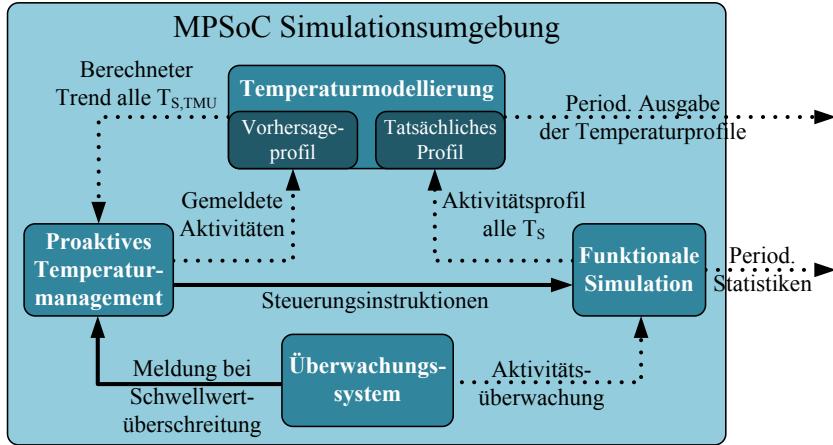


Abbildung 4.1: Schema der Umgebung zur Simulation des proaktiven Temperaturmanagements NoC-basierter Mehrkernprozessoren mithilfe des in Abschnitt 3.1 implementierten Temperaturmodells (Legende: → simulierte Datenübertragung im NoC, ⋯▷ simulatorinterne Kommunikation)

Proaktive TMU Die interne Funktionsweise der TMU ist detailliert in Abbildung 4.2 dargestellt. Die von den einzelnen Sonden des Überwachungssystems eingehenden Pakete werden zunächst analysiert und dem Vorhersageprofil übergeben. Dieses liefert alle $T_{S, TMU}$ den resultierenden Temperaturtrend, welcher in den TMU-internen Status sämtlicher Systemkomponenten einfließt. Neben der Temperatur beinhaltet dieser Status auch die aktuelle Betriebsfrequenz und für den Fall, dass es sich um einen IPC handelt, den derzeitigen Betriebsmodus. Hierbei wird zwischen Leerlauf und Last unterschieden.

Die Temperatur aller Komponenten wird periodisch abgefragt, wobei mehrere Temperaturschwellwerte geprüft werden. Falls einer dieser Schwellwerte über- oder unterschritten wird, werden unterschiedliche Instruktionen versendet. Wird beispielsweise ein über einen bestimmten Wert $T_{Schwell}$ hinausgehender Temperaturanstieg für eine Komponente prognostiziert, so wird versucht dessen Leistungsfähigkeit schrittweise durch eine Senkung der Betriebsfrequenz zu reduzieren bis kein weiterer Anstieg zu erwarten ist. Sobald das Absinken der Temperatur einer Komponente prognostiziert wird, wird ihre volle Leistungsfähigkeit schrittweise wiederhergestellt.

Handelt es sich bei der abgefragten Komponente um einen IPC, wird dieser darüber hinaus bei der Überschreitung einer maximal zulässigen Temperatur T_{Max} in einen Leerlaufzustand versetzt. Die auf dem IPC bearbeitete Aufgabe wird pausiert und durch die TMU auf den kühlstens IPC verschoben, um Temperaturspitzen auszugleichen. Ähnliches gilt auch für die maximal zulässige Temperaturdifferenz ΔT_{Max} . Diese wird genutzt, um das Temperaturprofil auszubalancieren. Sobald die größte ermittelte Temperaturdifferenz zwischen allen IPCs mindestens ΔT_{Max} beträgt, wird die auf dem heißesten IPC bearbeitete Aufgabe mit der auf dem kühilstens IPC bearbeiteten Aufgabe ausgetauscht.

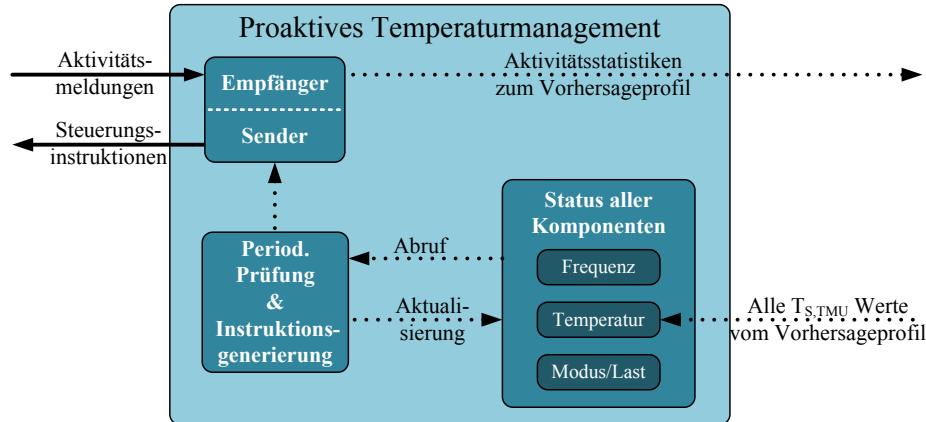


Abbildung 4.2: Interne Funktionsweise des proaktiven Temperaturmanagements: periodische Überprüfung der Temperatur aller Systemkomponenten

Dieser Ansatz entspricht im Wesentlichen dem ereignisbasierten Konzept herkömmlicher reaktiver Ansätze. Der Unterschied besteht darin, dass die Temperatur direkt in der TMU überwacht und geprüft wird. Die Prüfung der maximal zulässigen Temperaturänderung $T_{Schwelle}$ findet, unabhängig davon für welche Komponenten Aktivitätsmeldungen eingetroffen sind, immer für alle Komponenten des gesamten Systems statt. Dies ist einer von mehreren Unterschieden zu einem reaktiven Management, bei welchem nur die Prüfung der Temperatur der Komponente sinnvoll ist, für die ein aktueller Wert eingegangen ist. Weitere mögliche Änderungen in der Umgebung bleiben reaktiven Konzepten damit vorerst verborgen. Wird durch die proaktive TMU eine Instruktion generiert, wird diese über das NoC an die entsprechende Komponente gesendet. Weiterhin werden die resultierenden Statusänderungen im TMU-internen Statusregister vermerkt.

Wie bereits deutlich geworden ist, beschränken sich die Steuerungsanweisungen auf DFS sowie eine Umverteilung der Last, um lokal die Leistungsaufnahme und damit die Wärmeabstrahlung zu reduzieren. Die implementierten Konzepte stellen nicht unbedingt die effektivsten Maßnahmen dar. Beispielsweise wäre eine Anpassung der Betriebsspannung durch DVS mit einem quadratischen Einfluss auf die Leistungsaufnahme weitaus wirksamer. Sie genügen jedoch für die Überprüfung der Effizienz und Effektivität des proaktiven Temperaturmanagements sowie einen Vergleich mit einer reaktiven Variante. Zudem verursacht DVS, wie einige andere Methoden auch, Nebeneffekte (siehe Abschnitt 4.1), deren Modellierung die Simulation unnötig verkomplizieren würde.

Überwachungssystem Die Realisierung des Überwachungssystems beeinflusst auf entscheidende Weise die Effizienz des Systemmanagements. Daher werden im Rahmen des proaktiven Temperaturmanagements anstelle von Temperaturen die Schaltaktivitäten der einzelnen Komponenten überwacht, da eine Temperaturüberwachung zu unnötig vielen

Überwachungspaketen führt. Die Ursache hierfür liegt darin, dass auch sinkende Temperaturen und kurzzeitige Schwankungen gemeldet werden. Dies entfällt für ein proaktives Management, da lediglich Aktivitäten und die zugehörigen Zeitspannen überwacht werden. Eine reine Aktivitätsüberwachung führt somit theoretisch zu einer Reduzierung des Überwachungsverkehrs und dadurch zu einer geringeren zusätzlichen Belastung. Dies setzt allerdings voraus, dass Aktivitätsmeldungen weniger oft als Temperaturmeldungen generiert werden.

Des Weiteren ist das Überwachungssystem nicht ausschließlich auf Temperatursensoren angewiesen und setzt stattdessen auf wesentlich weniger komplexe Aktivitätszähler. Diese benötigen im Gegensatz zu Temperatursensoren keine energieintensive und mit Verzögerungen verbundene Umwandlung der analogen Messwerte in digital übertragbare Daten, sondern können die digitalen Zählwerte direkt übertragen. Herkömmliche Temperatursensoren können allerdings nicht vollständig entfallen. Diese werden nach wie vor benötigt, um Temperatureinflüsse außerhalb des Vorhersagemodells (z. B. schwankende Umgebungstemperaturen) berücksichtigen zu können oder Mechanismen für die Kalibrierung und Korrektur des Profilverlaufs zu ermöglichen. Allerdings können Häufigkeit und Dauer der Inanspruchnahme der Temperatursensoren deutlich reduziert werden. Auf diese Weise lässt sich die Anzahl der analogen Messungen und der dazugehörigen Analog-Digital-Wandlungen reduzieren. Im Vergleich dazu wird davon ausgegangen, dass die durch die Aktivitätszähler verursachte Leistungsaufnahme vernachlässigbar gering ist.

Die angesprochenen Mechanismen für Kalibrierung und Korrektur werden im Rahmen dieser Arbeit nicht weiter betrachtet, da einerseits alle zu kalibrierenden Parameter (z. B. Umgebungstemperatur und anfängliche Chiptemperatur) durch die Simulationsumgebung bereitgestellt werden. Andererseits verfälschen Korrekturmaßnahmen die Auswirkungen eines rein proaktiven Temperaturmanagements auf die Leistungsfähigkeit und die Temperaturverteilung eines Systems. Der Verzicht auf Korrekturen ist zwar keinesfalls realitätsnah, erhöht jedoch die Sichtbarkeit der Auswirkungen und verdeutlicht damit die Charakteristika eines proaktiven Ansatzes. Im Rahmen realistischer Anwendungsszenarien sind sensorgestützte Kalibrierungs- und Korrekturmaßnahmen selbstverständlich unerlässlich.

Die prinzipielle Organisation des Überwachungssystems ist in Abbildung 4.3 dargestellt. Zunächst wird jeder Komponente ein, als Messpunkt „M“ bezeichneter, Aktivitätszähler zugeordnet. Für die Aktivitätszählung bieten sich sowohl die Zählung einzelner Bitübergänge als auch die Zählung eintreffender Flits an. Da mithilfe der ersten Möglichkeit genauere Abschätzungen bezüglich der aufgenommenen Leistung möglich sind, wird diese im Folgenden verwendet.

Würde jeder Messpunkt seine registrierten Bitübergänge selbstständig an die TMU

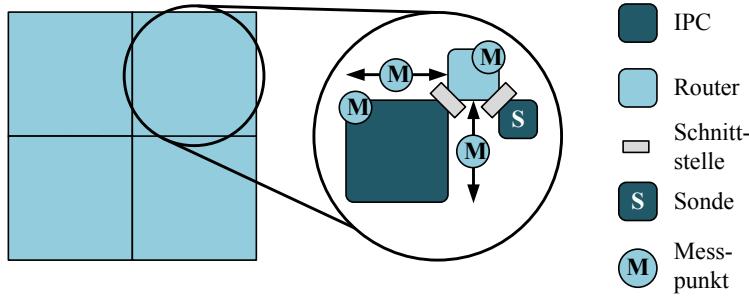


Abbildung 4.3: Einteilung eines 2×2 NoC in Zuständigkeitsbereiche: jede Sonde überwacht die Aktivität des ihr zugewiesenen Routers, des IPCs und der Übertragungskanäle mithilfe einzelner Messpunkte

weiterleiten, würde dies vermutlich ein sehr hohes Verkehrsaufkommen hervorrufen. Daher wird das NoC in Zuständigkeitsbereiche eingeteilt. Zu einem solchen Bereich gehören ein IPC, der dazugehörige Router sowie jeweils ein Übertragungskanal in Ost-West- und Nord-Süd-Richtung. Wie aus der Abbildung hervorgeht, ergibt dies vier Messpunkte, welche einer Sonde unterstellt sind. Die Sonde selbst ist dafür verantwortlich die Überschreitung des Aktivitätsschwellwertes $Akt_{Schwell}$ durch einen Messpunkt samt verstrichener Zeit festzuhalten, diese Werte für eine Übertragung an die TMU aufzubereiten und den Messpunkt zurückzusetzen. In diesem Beispiel erfolgt die Anbindung der Sonde an das Kommunikationsnetzwerk durch einen zusätzlichen Router-Port. Weitere Anbindungsmöglichkeiten werden im späteren Verlauf diskutiert.

Die eigentlichen Messpunkte können beispielsweise Zähler sein, welche in den NoC-Komponenten und IPCs bereits als Hardware vorhanden sind. Für die Bestimmung der Aktivität der IPCs wäre auch eine Abschätzung mithilfe charakteristischer Lastfaktoren oder die Ausnutzung von außen zugänglicher Statusregister oder Performance Counter [AV02] denkbar. Dies ist besonders für IPCs von Drittanbietern relevant, da für diese ein Zugriff auf interne Funktionalitäten oder Komponenten oftmals untersagt ist.

Für die Übertragung der erhobenen Statistiken von den Sonden zur TMU ist im Rahmen dieser Arbeit die Nutzung der vorhandenen NoC-Infrastruktur vorgesehen. Mithilfe dieser werden sowohl die Überwachungsdaten als auch die Steuerungsinstruktionen genauso wie regulärer Datenverkehr übertragen. Die Pakete, welche die Überwachungsdaten beziehungsweise Steuerungsanweisungen enthalten, sind mit entsprechenden Bits im Paket-Header gekennzeichnet. Darüber hinaus haben alle Sonden des Überwachungssystems jederzeit Kenntnis von der Position der TMU im NoC. Dies ist von Bedeutung, da es im Verlauf des Systemmanagements zu einer Verlagerung dieser kommen kann. Weitere Möglichkeiten wären die Verwendung dedizierter Steuerleitungen und -signale der NoC-Komponenten oder die Ausnutzung von Teststrukturen und -mechanismen (z. B. Built-In-Self-Test, Boundary-Scan-Test) [TDB⁺06]. Auch die Realisie-

rung eines redundanten NoC für Steuer- und Überwachungsdaten stellt eine Option dar, erscheint aber aufgrund des hohen Platzbedarfs und der zusätzlichen Leistungsaufnahme verglichen zum erzielten Mehrwert wenig vielversprechend [CGRB06].

Integration Da für die Übertragung des Überwachungs- und Steuerungsverkehrs das vorhandene NoC genutzt werden soll, ist eine effiziente Integration der TMU und der Sonden unumgänglich. Drei nennenswerte Varianten der Anbindung der Sonden in das NoC sind in Abbildung 4.4 illustriert.

Die Variante in Abbildung 4.4(a) sieht eine Integration der Sonde in den IPC des Zuständigkeitsbereichs der Sonde vor. Damit teilen sich Sonde und IPC dieselbe Schnittstelle, sodass eine parallele Datenübertragung beider Komponenten nicht möglich ist. Diese Variante der Integration verursacht außer für die Sonde selbst weder Hardware-Kosten noch eine Erhöhung der Leistungsaufnahme [C.11]. Sie hat allerdings eine gegenseitige Behinderung von IPC und Sonde zur Folge. Darüber hinaus ist die nötige Modifikation des IPC durch den geistigen Eigentümer möglicherweise untersagt. Zudem verstößt diese Variante gegen das Modularitätsprinzip NoC-basierter Systeme.

Für die in Abbildung 4.4(b) dargestellte Variante wird die Sonde außerhalb des IPC platziert. Dadurch wird ein Betrieb der Sonde auch dann ermöglicht, wenn der IPC nicht verfügbar ist. Beide Komponenten teilen sich jedoch nach wie vor einen Port des Routers. Daher ist eine geteilte Schnittstelle notwendig, welche zwischen IPC und Sonde umschaltet und deren Sendeanfragen koordiniert. Die zusätzliche Schnittstelle bedingt eine Erhöhung der Hardware-Kosten und der Leistungsaufnahme [C.11]. Durch die Nutzung desselben Ports ist auch für diese Variante keine parallele Datenübertragung möglich. Darüber hinaus erhöht sich die Dauer der Datenübertragung durch die Schnittstelle.

Abbildung 4.4(c) illustriert die Variante mit den höchsten Kosten, da die Sonde über einen zusätzlichen Router-Port mit dem Netzwerk verbunden wird. Dies erhöht nicht nur die verursachten Hardware-Kosten [C.11], sondern steigert auch die Komplexität der übrigen Ports. Die Folge ist eine erhöhte Leistungsaufnahme des Gesamtsystems. Zusätzlich verringert sich potentiell auch die Leistungsfähigkeit, da die Router eine höhere Anzahl von Ports arbitrieren müssen. Somit sinkt die Häufigkeit der Paketübertragung durch einen einzelnen Port. Im Gegenzug ist die Sonde unabhängig vom IPC an das Netz angebunden, sodass eine getrennte Einspeisung von regulären Daten und Überwachungsdaten in das Netz erfolgen kann. Damit entfällt auch die bei der zweiten Variante entstehende, zusätzliche Latenz der Datenübertragung. Zudem bleibt die Modularität erhalten und die Sonde ist auch bei Nichtverfügbarkeit des IPC betriebsfähig.

Eine Zusammenfassung der Vor- und Nachteile der einzelnen Anbindungsalternativen wird in Tabelle 4.1 gegeben. Besonders aufgrund der Einhaltung des Modularitätsprinzips NoC-basierter Systeme sowie der unkomplizierten Realisierbarkeit eignet sich

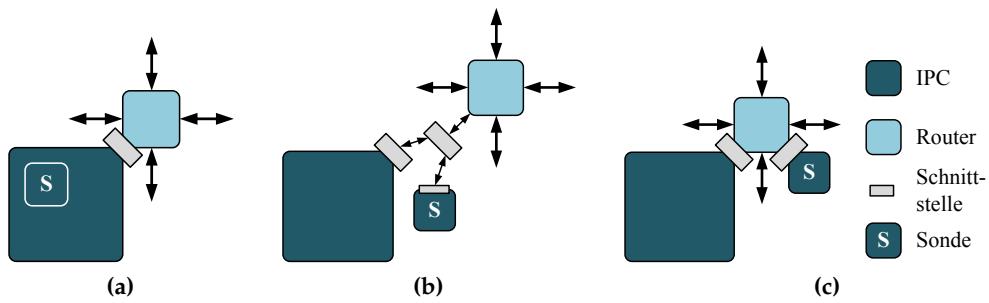


Abbildung 4.4: Alternativen eine Sonde an das NoC anzuschließen: (a) Integration in einen IPC, (b) zusätzliche, geteilte Schnittstelle, (c) zusätzlicher Router-Port

die Anbindung der Sonde mithilfe eines zusätzlichen Router-Ports am besten für den weiteren Verlauf dieser Arbeit. Zwar wird die Leistungsfähigkeit des Systems reduziert. Dies gilt jedoch auch für ein entsprechendes zu Vergleichszwecken herangezogenes, reaktives Temperaturmanagement. Abschätzungen bezüglich des Implementierungs- und Integrationsaufwands ähnlicher Sonden sind [WCTT10] zu entnehmen.

Für ein leistungsfähiges proaktives Temperaturmanagement ist auch die Integration der TMU in das von ihr zu steuernde NoC-basierte System von großer Bedeutung. Im Rahmen dieser Arbeit wird die TMU in einen, sich im NoC befindlichen, IPC integriert. Dabei ergänzt die TMU die Funktionalität des ursprünglichen IPC. Solange keine internen Funktionen der TMU ausgeführt oder Steuerungsinstruktionen generiert werden müssen, arbeitet der IPC wie vorgesehen. Andernfalls haben die Aufgaben des Temperaturmanagements Vorrang und der reguläre Betrieb wird pausiert.

Da Steuerungsinstruktionen existieren, welche eine Umverteilung der IPC-Last vorsehen, muss dies bei der Integration der TMU berücksichtigt werden. Diese ist von der Umverteilung der Last nicht ausgeschlossen, sodass die TMU je nach Temperaturverlauf möglicherweise mehrfach verlagert wird. Eine Verlagerung kann durch eine erhöhte Leistungsaufnahme sowie eine gesteigerte Aktivität in der Umgebung der TMU notwendig werden. Ersteres wird durch die Aktivität der TMU hervorgerufen, Letzteres wird durch eintreffende Überwachungsdaten verursacht. Diese sorgen für zusätzliche Wärmeabstrahlung und machen unter Umständen eine Verlagerung erforderlich. Damit ist im Grunde jeder sich im NoC befindliche IPC eine potentielle TMU.

Die vorgestellte Art und Weise der Integration der TMU lässt für die Anwendung des proaktiven Temperaturmanagements für einen realen NoC-basierten Mehrkernprozessor nur eine software-basierte Lösung zu. Da die Simulationsumgebung und das darin integrierte, proaktive Temperaturmanagementsystem als SystemC-basierte Software implementiert sind, erscheint dies allerdings ohnehin als der günstigste Ansatz. Somit könnte das proaktive Temperaturmanagement auf einem der Rechenkerne ausgeführt werden und bei Bedarf auf andere Kerne migriert werden [BBWW07].

Tabelle 4.1: Qualitative Bewertung der Anbindungsalternativen einer Sonde bezüglich der Hardware-Kosten, der zusätzlichen Leistungsaufnahme, der Möglichkeit der parallelen Datenübertragung durch Sonde und IPC und der Wahrung der IPC-Modularität (Legende: + gut, \circ neutral, - schlecht)

	Integriert	Geteilte Schnittstelle	Zusätzl. Port
Kosten	+	\circ	-
Leistungsaufnahme	+	\circ	-
Parallelität	-	-	+
Modularität	-	+	+

Reaktives Temperaturmanagement

Das zu Vergleichszwecken implementierte reaktive Temperaturmanagement ähnelt dem proaktiven Konzept in vielen Punkten. So greift beispielsweise auch der reaktive Ansatz auf eine ereignisgesteuerte Überwachung NoC-basierter Systeme zurück [CBR⁺05]. Der Unterschied liegt darin, dass die Sonden anstelle von Aktivitäten die mithilfe der Messpunkte ermittelten Temperaturwerte der einzelnen NoC-Komponenten überwachen. Der generelle Ablauf des reaktiven Temperaturmanagements ist in Abbildung 4.5 dargestellt. Im Gegensatz zum proaktiven Konzept existiert aber lediglich ein Temperaturmodell, welches als „Tatsächliches Profil“ gekennzeichnet ist. Dieses Modell repräsentiert den tatsächlichen Temperaturverlauf und wird von den Sonden des Überwachungssystems genutzt. Die Sonden benachrichtigen die reaktive TMU, sobald Temperaturänderungen größer als $T_{Schwell}$ zu verzeichnen sind. Diese reagiert mit entsprechenden Maßnahmen.

Reaktive TMU Auch die Funktionsweise der reaktiven TMU (siehe Abbildung 4.6) unterscheidet sich nur in wenigen Punkten von der proaktiven Variante. Entscheidend ist, dass die reaktive TMU nur Zugriff auf die durch das Überwachungssystem gemeldeten Temperaturverläufe hat und nicht periodisch einen aktualisierten Trend des vollständigen Temperaturprofils erhält. Auf Basis der Informationen, welche der TMU zur Verfügung stehen, versendet diese gegebenenfalls Instruktionen an die betroffenen NoC-Komponenten und vermerkt dies in ihrem internen Statusregister. Die Prüfung, ob Maßnahmen eingeleitet werden müssen, findet nicht periodisch und nicht für sämtliche Komponenten statt.

Stattdessen wird die Prüfung durch das Eintreffen einer Überwachungsnachricht ausgelöst und wird nur für die betroffene Komponente bezüglich der Werte T_{Max} und ΔT_{Max} durchgeführt. Die Werte $Akt_{Schwell}$ und $T_{S,TMU}$ haben für das reaktive Tempe-

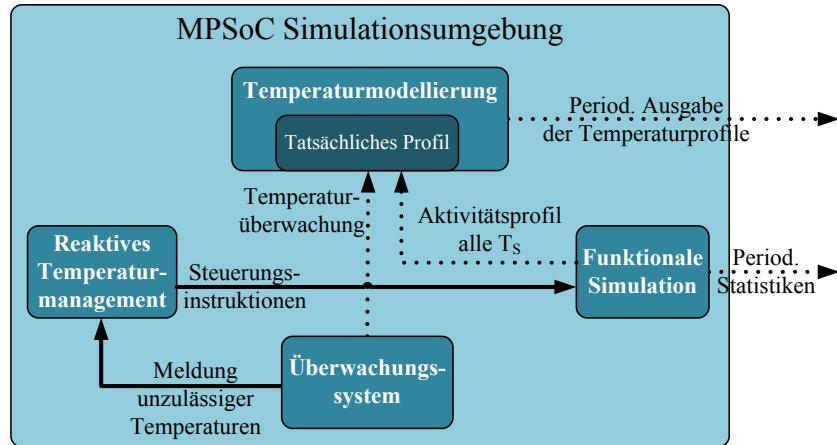


Abbildung 4.5: Schema der Umgebung zur Simulation des reaktiven Temperaturmanagements NoC-basierter Mehrkernprozessoren mithilfe des in Abschnitt 3.1 implementierten Temperaturmodells (Legende: → simulierte Datenübertragung im NoC, ⋯▷ simulatorinterne Kommunikation)

raturmanagement keinerlei Bedeutung. Die Instruktionen selbst beschränken sich auch für die reaktive TMU auf DFS und eine Umverteilung der Last. Die reaktive TMU wird ebenfalls in einen vorhandenen IPC integriert. Solange keine Überwachungsnachrichten eintreffen, befindet sich die TMU im Leerlauf und der IPC arbeitet wie vorgesehen. Je nach Häufigkeit eintreffender Meldungen begünstigt dies im Vergleich zum proaktiven Konzept die Leistungsfähigkeit des IPC und reduziert die Leistungsaufnahme der TMU.

Überwachungssystem und Integration Die Sonden des Überwachungssystems des reaktiven Temperaturmanagements sind mithilfe eines zusätzlichen Router-Ports (siehe Abbildung 4.4(c)) an das Kommunikationsnetzwerk angebunden und nutzen ebenfalls Messpunkte zur Überwachung. Analog zum proaktiven Konzept wird das NoC in Zuständigkeitsbereiche eingeteilt (siehe Abbildung 4.3). Jede Sonde überwacht auch hier einen IPC, den zugehörigen Router sowie die beiden Übertragungskanäle.

Im Unterschied zum proaktiven Überwachungssystem werden allerdings Temperaturwerte überwacht, was die bereits erwähnten Nachteile einer verzögernden und energieintensiven Analog-Digital-Wandlung mit sich bringt. Darüber hinaus müssen auch Temperatursenkungen gemeldet werden, da die TMU nur auf diese Weise von einer Entspannung der Lage Kenntnis erlangen kann und die volle Leistungsfähigkeit einzelner Komponenten wiederherstellen kann. Die eigentliche Temperaturmessung ist ereignisgesteuert. Wird für einen der Messpunkte eine Temperaturänderung detektiert, welche über einen vorgegebenen Schwellwert hinausgeht, so übermittelt die jeweilige Sonde die Temperatur der betroffenen Komponente an die TMU.

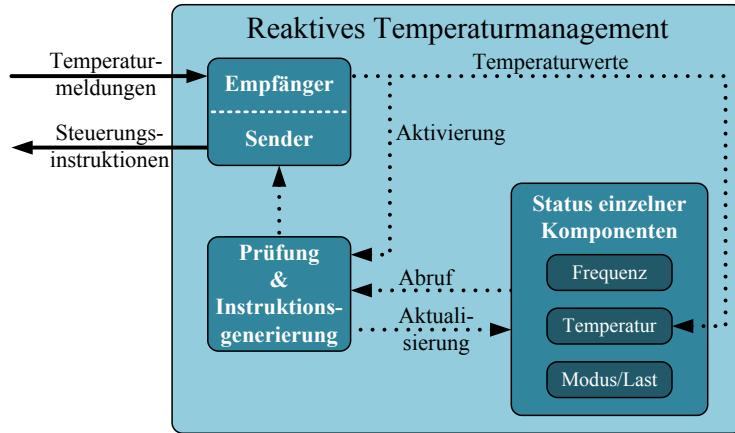


Abbildung 4.6: Interne Funktionsweise des reaktiven Temperaturmanagements: der Status einzelner Systemkomponenten wird überprüft, sobald eine Temperaturmeldung für diese eintrifft

Entscheidende Unterschiede zwischen den Konzepten

Der entscheidende Vorteil des proaktiven Temperaturmanagements gegenüber dem reaktiven Konzept liegt in der verkürzten Antwortzeit auf detektierte unzulässige Temperaturwerte. Die Antwortzeit setzt sich generell aus der Dauer für das Detektieren eines Ereignisses, die Benachrichtigung der TMU, die Verarbeitung durch die TMU sowie das Übermitteln entsprechender Steuerungsinstruktionen an die Komponente zusammen. Dieser Ablauf ist in Abbildung 4.7 beispielhaft für einen Router in einem 3×3 NoC dargestellt.

Antwortzeit des reaktiven Managements Im Rahmen des reaktiven Temperaturmanagements wird dabei ein unzulässiger Temperaturanstieg des Routers durch die zuständige Sonde detektiert. Dies wird der TMU mithilfe einer Nachricht mitgeteilt, welche über das NoC übertragen wird. Anschließend wird durch die TMU eine Instruktion generiert und zum Router gesendet. Die Antwortzeit $\Delta t_{Reaktiv}$ setzt sich somit im Wesentlichen aus $\Delta t_{Ausbreitung}$ und zweimal $\Delta t_{Übertragung}$ zusammen (siehe Gleichung 4.1). $\Delta t_{Übertragung}$ ist die Zeit, welche die Überwachungs- und Instruktionsnachrichten für ihren Weg durch das NoC benötigen. Neben der Entfernung zwischen Sonde und TMU hängt diese Dauer stark von der Lastsituation im NoC ab. $\Delta t_{Ausbreitung}$ repräsentiert die Zeit, welche zwischen dem Auftreten einer Schaltaktivität und einer entsprechenden messbaren Wärmeausbreitung vergeht. Diese Zeit ist von der thermischen Zeitkonstante τ_{th} abhängig und beschreibt die für eine Erwärmung oder Abkühlung eines Körpers nötige Dauer (siehe Abschnitt 3.1.1). τ_{th} ergibt sich aus dem Produkt des thermischen Widerstandes R_{th} und der thermischen Kapazität C_{th} (siehe Gleichung 4.2) und hängt von den Ausmaßen des betrachteten Körpers sowie dessen Materialeigenschaften ab.

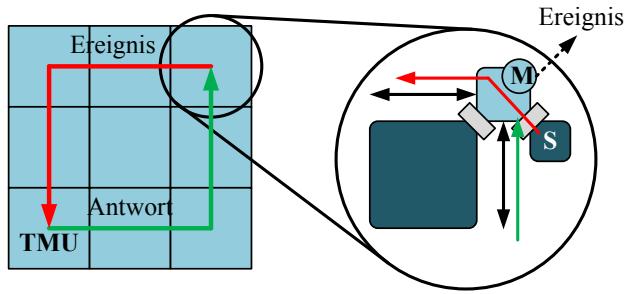


Abbildung 4.7: Exemplarische Kommunikation zwischen Sonde und TMU: die Sonde detektiert an einem der Messpunkte ein Ereignis und meldet dies der TMU, nach Erhalt der Nachricht übermittelt diese Maßnahmen an die betroffene Komponente

In Abschnitt 2.4.1 konnte bereits gezeigt werden, dass der durch die Leistungsaufnahme einer Schaltung verursachte Temperaturanstieg und die Ausbreitung der Wärme in umliegende Regionen mehrere hundert Millisekunden in Anspruch nehmen. Damit kann $\Delta t_{\text{Ausbreitung}}$ vergleichsweise große Werte annehmen. Dies schlägt sich wiederum in langen Antwortzeiten des reaktiven Managements auf Temperaturänderungen nieder. Einerseits führt dies zu verzögerten Reaktionen auf kritische Temperaturwerte und unnötig langen Phasen verringriger Leistungsfähigkeit der Komponenten, da auch Abkühlungen erst der TMU gemeldet werden müssen, bevor der Normalbetrieb wiederhergestellt werden kann. Andererseits sorgt die Notwendigkeit der Meldung von Abkühlungen für zusätzlichen Verkehr im NoC, welcher den regulären Datenverkehr behindert und möglicherweise $\Delta t_{\text{Reaktiv}}$ beeinträchtigt.

$$\Delta t_{\text{Reaktiv}} = \Delta t_{\text{Ausbreitung}}(\tau_{th}) + 2 \cdot \Delta t_{\text{Übertragung}} \quad (4.1)$$

$$\tau_{th} = R_{th} \cdot C_{th} \quad (4.2)$$

Antwortzeit des proaktiven Managements Für das Beispiel des Routers in Abbildung 4.7 überwacht die Sonde des proaktiven Konzepts im Gegensatz dazu die Aktivität des Routers. Dies hat unter anderem den Vorteil, dass durch das Entfallen einer Analog-Digital-Wandlung Energie gespart werden kann, da anstatt analoger Messwerte digitale Zählwerte ausgewertet werden. Die Leistungsaufnahme eines einzelnen Analog-Digital-Wandlers, wie er auch für die Temperaturdiode des in Abschnitt 2.4.1 genutzten Virtex5-FPGAs eingesetzt wird, beträgt in der Regel mehrere Milliwatt [LRRB05, SMS09].

Die Aktivitätsstatistiken werden der TMU über das NoC mitgeteilt. Diese lässt die bereitgestellten Informationen in ihr internes Temperaturmodell des Systems einfließen und prognostiziert beispielsweise einen Temperaturanstieg. Daraufhin wird eine entsprechende Instruktion generiert und zum Router gesendet, sodass dieser geeignete Maß-

nahmen einleiten kann. Die Antwortzeit $\Delta t_{Proaktiv}$ setzt sich damit aus $\Delta t_{Berechnung}$ und ebenfalls zweimal $\Delta t_{Übertragung}$ zusammen (siehe Gleichung 4.3). $\Delta t_{Übertragung}$ ist identisch zu dem für das reaktive Konzept vorgestellten Wert. Auch hier müssen Überwachungsstatistiken und Instruktionen übertragen werden.

Der entscheidende Unterschied zwischen den Konzepten liegt in der Dauer von $\Delta t_{Berechnung}$. Diese beschreibt im Gegensatz zu $\Delta t_{Ausbreitung}(\tau_{th})$ die Dauer, welche für die Berechnung der Auswirkungen aktuell in der TMU eingetroffener Aktivitätsstatistiken auf den Temperaturtrend notwendig ist. $\Delta t_{Berechnung}$ ist damit unabhängig von physikalisch festgelegten Größen wie τ_{th} und hängt vielmehr direkt von der Genauigkeit des verwendeten Temperaturmodells ab. Modelle höherer Auflösung sorgen zwar für genauere Temperaturabschätzungen, durch den erhöhten Berechnungsaufwand verlängert sich allerdings auch $\Delta t_{Berechnung}$. Um zu gewährleisten, dass $\Delta t_{Antwort}$ für das proaktive Management kürzer als für das reaktive Gegenstück ist, muss das Temperaturmodell entsprechend gewählt werden. $\Delta t_{Berechnung}$ muss dabei kleiner als $\Delta t_{Ausbreitung}$ sein, während die prognostizierten Werte trotzdem ausreichend genau sind. Dass dies durchaus möglich ist, wurde in Abschnitt 3.2 gezeigt. Kurze Werte für $\Delta t_{Berechnung}$ tragen darüber hinaus zu einer Reduzierung der durch die TMU zusätzlich erzeugten Abwärme bei.

Weiterhin wird durch die Überwachung von Schaltaktivitäten der Überwachungsverkehr reduziert, da eine Temperaturverringerung automatisch durch das Temperaturmodell berechnet wird. Dies verringert sowohl die Verkehrslast im Netzwerk als auch die durch die Überwachungsnachrichten verursachte Abwärme. Selbstverständlich trifft dies nur dann zu, wenn der Schwellwert für das Detektieren von Schaltaktivitäten so gewählt wird, dass für das proaktive Konzept weniger Überwachungsnachrichten als für das reaktive Konzept erzeugt werden. Die Annahme, dass die Zeitersparnis für $\Delta t_{Proaktiv}$ tatsächlich relevant ist, wird durch vergleichsweise niedrige Werte von einigen Nanosekunden für $\Delta t_{Übertragung}$ untermauert (siehe [WGT12] für die Übertragungsdauer D_P eines regulären Datenpaketes).

$$\Delta t_{Proaktiv} = \Delta t_{Berechnung} + 2 \cdot \Delta t_{Übertragung} \quad (4.3)$$

Besonders aufgrund verkürzter Antwortzeiten ermöglicht das proaktive Management mehrere Vorteile gegenüber reaktiven Ansätzen. Vorausgesetzt die Temperaturprofile der zugrunde liegenden Systeme können positiv beeinflusst werden, ergeben sich zwei Varianten dieses Vorteil ausnutzen.

Entweder können unter Nutzung identischer Steuerungsparameter Temperaturspitzen und thermischer Stress durch den Zeitvorteil reduziert werden. Oder die Kriterien für den Einsatz der Steuerungsmaßnahmen können gelockert werden, um im Vergleich mit reaktiven Konzepten gleiche Ergebnisse zu erzielen und dabei die Leistungsfähigkeit des Systems in geringerem Maß zu beeinträchtigen.

Ob und inwieweit die getroffenen Aussagen bezüglich des Zeit- und Lastvorteils des proaktiven Managements gegenüber reaktiven Methoden tatsächlich zutreffen, wird im folgenden Abschnitt mithilfe einer Simulationsreihe überprüft. Dabei werden gleichermaßen leistungs- als auch temperaturbezogene Aspekte berücksichtigt.

4.2.2 Vergleich mit reaktivem Management und Bewertung

In diesem Abschnitt wird der vorgestellte Ansatz für ein proaktives Temperaturmanagement bezüglich seines Einflusses auf die Temperatur sowie die Leistungsfähigkeit NoC-basierter Mehrkernprozessoren untersucht. Zu diesem Zweck wird das proaktive Management mit dem vorgestellten reaktiven Managementkonzept und mit einem Referenzsystem ohne Temperaturmanagement verglichen. Die Betrachtungen konzentrieren sich auf das Vermögen der beiden Ansätze, das Temperaturprofil der zugrunde liegenden Systeme positiv beeinflussen zu können sowie auf eventuelle Beeinträchtigungen leistungsbezogener Parameter.

Sämtliche Simulationen werden für ein 4×4 NoC durchgeführt. Einerseits soll damit gewährleistet werden, dass ein NoC mit repräsentativen Dimensionen betrachtet wird. Andererseits muss das NoC klein genug gewählt werden, damit die Effekte der Managementstrategien möglichst deutlich sichtbar werden. Dies ist potentiell eher für kleinere Netzwerke der Fall, da diese eine höhere Auslastung aufweisen und das Management somit weniger Entscheidungsspielraum hat. Schwachstellen bezüglich der Überwachung oder getroffener Managemententscheidungen können damit wesentlich einfacher identifiziert werden. Die Simulationsparameter sind identisch zu den in Tabelle 3.3 aufgeführten Einstellungen. Zusätzlich sind in Tabelle 4.2 verschiedene Konfigurationen für das reaktive und das proaktive Temperaturmanagement aufgelistet.

Die Konfiguration K1.1 dient als Standardkonfiguration. Für die Konfigurationen K1.2 und K1.3 werden die Auswirkungen einer geänderten maximal zulässigen Temperaturänderung $T_{Schwell}$ untersucht. Für das proaktive Management bedeutet dies, dass mit größeren Werten prognostizierte Temperaturänderungen erst später als unzulässig eingestuft werden. Für das reaktive Management melden die Sonden unzulässige Temperaturen später. Die Konfigurationen K2.1 und K2.2 dienen der Überprüfung der Effekte verschiedener Werte für $Akt_{Schwell}$. Dieser Parameter beeinflusst lediglich das proaktive Management und reguliert die Häufigkeit der Aktivitätsmeldungen durch die Sonden an die TMU. Auch die TMU-interne Simulationsperiode $T_{S,TMU}$ ist nur für das proaktive Management relevant und wird mithilfe der Konfigurationen K3.1 und K3.2 untersucht. Höhere Werte für $T_{S,TMU}$ sorgen für eine niedrigere Frequenz, mit der die TMU aktualisierte Temperaturwerte erhält. Die Auswirkungen verschiedener Werte für T_{Max} und ΔT_{Max} werden nicht untersucht, da diese für beide Ansätze die Verlagerung von auf den IPCs bearbeiteten Aufgaben auf identische Weise beeinflussen. Der wesentlich be-

Tabelle 4.2: Parameterkonfigurationen für die Simulation des proaktiven und reaktiven Temperaturmanagements

	$T_{Schwell}[^{\circ}\text{C}]$	$Akt_{Schwell}$	$T_{S, TMU}[\text{ms}]$
K1.1	1	100 000	1
K1.2	2	100 000	1
K1.3	4	100 000	1
K2.1	1	50 000	1
K2.2	1	200 000	1
K3.1	1	100 000	2
K3.2	1	100 000	5

deutsamere Mechanismus der Detektierung und weiteren Behandlung unerwünschter Temperaturtrends bleibt davon unberührt.

Temperatur

Entscheidend für die Güte einer Strategie für ein Temperaturmanagement ist, ob und inwieweit dieses die Temperatur des zugrunde liegenden Systems positiv beeinflussen kann. Im Folgenden werden die durchschnittliche sowie die maximale Temperatur im NoC betrachtet, um diesbezüglich Aussagen treffen zu können. Zusätzlich wird der maximale Temperaturunterschied zwischen den Systemkomponenten berücksichtigt. Ein Vergleich des Temperaturverlaufs einzelner Komponenten ist an dieser Stelle nicht sinnvoll durchführbar, da eventuell eingesetzte Steuerungsmaßnahmen einen fairen Vergleich erschweren. So gibt beispielsweise die Betrachtung des Temperaturverlaufs eines einzelnen Routers zwar einen gewissen Aufschluss über die Arbeitsweise der unterschiedlichen Strategien, doch sagt dieser Verlauf wenig über die Temperaturverteilung im restlichen System aus.

Die Ergebnisse für die untersuchten Temperaturparameter sind für die in Tabelle 4.2 aufgeführten Konfigurationen in Abbildung 4.8 zusammengefasst. Die Resultate stellen den Durchschnitt der Einzelwerte dar, welche über die gesamte Simulationsdauer aufgenommen wurden.

Durchschnittliche Temperatur Die durchschnittliche Temperatur im NoC kann bezogen auf das Referenzsystem durch ein proaktives Management für fast alle Konfigurationen um mindestens 1 °C reduziert werden. Gemäß der in Abschnitt 2.2.3 eingeführ-

ten Faustregel einer Halbierung der Lebensdauer hochintegrierter Schaltungen bei einer dauerhaften Temperaturerhöhung um 10 K, würde dies eine um 7 % erhöhte Lebensdauer gegenüber der Referenz bedeuten. Die für ein reaktives Management relevanten Konfigurationen bewirken hingegen keine oder nur geringfügige Reduzierungen. Im besten Fall (d. h. K2.1) verringert sich die durchschnittliche Temperatur durch ein proaktives Management um 3,1 °C, was einer Verlängerung der Lebensdauer um 24 % entspräche. Dies ist mit der häufigeren Weiterleitung der durch die Sonden registrierten Aktivitäten an die TMU zu erklären, welche durch einen niedrigeren Schwellwert verursacht wird. Die TMU ist somit in der Lage, Aktivitäten frühzeitiger in die Berechnung der Prognose einzubeziehen. Umgekehrt wird die Weiterleitung der Aktivitätsstatistiken durch hohe Werte für $Akt_{Schwell}$ verzögert, was eine Verschlechterung der Resultate bewirkt. Steigende Werte für $T_{Schwell}$ bewirken eine Erhöhung der Durchschnittstemperatur. Dies ist auf die verzögerte Einleitung temperaturreduzierender Maßnahmen zurückzuführen, sodass ein Temperaturanstieg nicht mehr vermieden werden kann.

Eine Erhöhung beziehungsweise Reduzierung von $T_{S,TMU}$ hat die erwarteten negativen und positiven Auswirkungen. Kleine Werte sorgen dafür, dass das TMU-interne Temperaturprofil häufiger aktualisiert wird und somit früher auf unerwünschte Trends reagiert werden kann. Dies erhöht allerdings den Berechnungsaufwand und trägt somit zusätzlich zur Generierung von Wärme bei. Hohe Werte hingegen verlängern die Zeiträume zwischen den Aktualisierungen und verbergen sich abzeichnende Temperaturtrends möglicherweise unnötig lange. Im Gegenzug reduziert sich dadurch der Berechnungsaufwand. Dieses Verhalten wird für die Maximaltemperatur sowie den maximalen Temperaturunterschied deutlicher sichtbar.

Maximaltemperatur und maximaler Temperaturunterschied Ähnliche Resultate werden für die maximal im NoC aufgetretene Temperatur erzielt. Auch hier erreicht das reaktive Management keine Reduzierung, sondern verursacht teilweise sogar eine Erhöhung, welche mit kleineren Werten für $T_{Schwell}$ zunimmt. Dies ist durch die größere Anzahl von Überwachungsnachrichten zu erklären, welche das Netzwerk durchlaufen und damit zusätzlich Verkehr und Wärme erzeugen.

Die proaktive Strategie verhält sich dazu entgegengesetzt, sodass sich die Maximaltemperatur mit steigenden Werten für $T_{Schwell}$ deutlich erhöht. Dies ist mit verzögerten Maßnahmen für eine Temperaturreduzierung zu erklären. Niedrige Werte für $Akt_{Schwell}$ und $T_{S,TMU}$ sorgen für eine Temperaturreduzierung, während hohe Werte das Gegenteil bewirken. K2.1 sorgt für die deutlichste Reduzierung der Maximaltemperatur um 6,6 °C. Dies entspräche einer Erhöhung der Lebensdauer um 58 % gegenüber der Referenz. Die charakteristische Ausprägung der Resultate für die durchschnittliche Temperatur sowie die Maximaltemperatur kann auch auf den maximalen Temperaturunterschied übertragen werden.

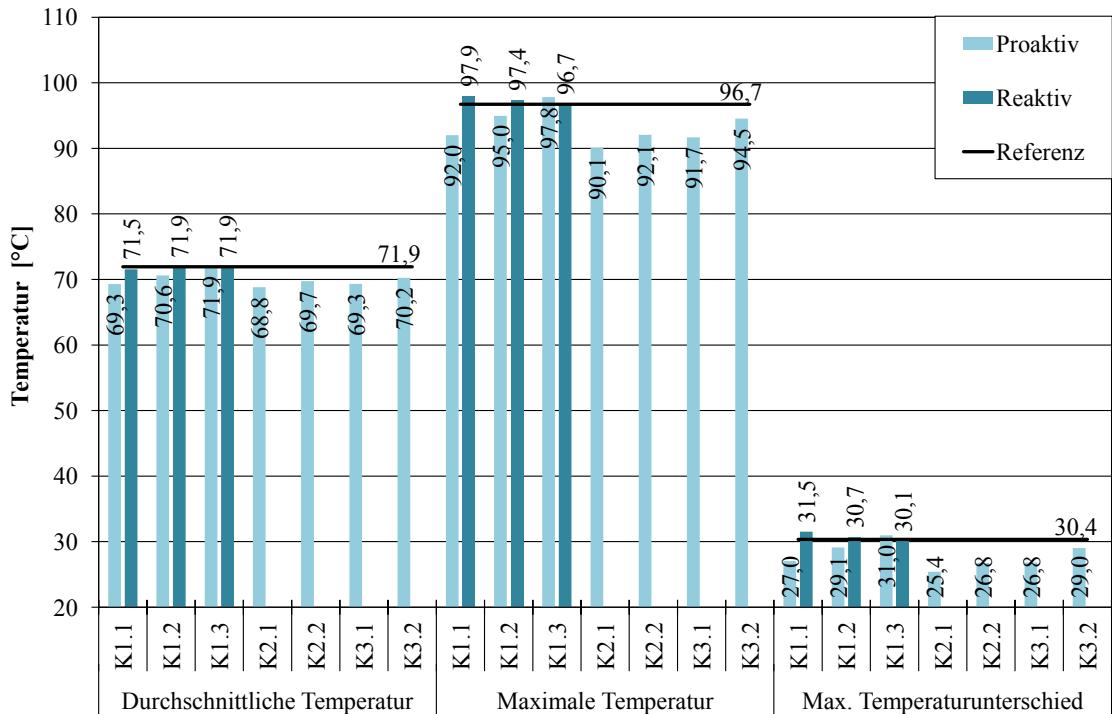


Abbildung 4.8: Durchschnittstemperatur, Maximaltemperatur und maximaler Temperaturunterschied in einem 4×4 NoC unter Verwendung verschiedener Konfigurationen für das proaktive sowie das reaktive Temperaturmanagement

gen werden. Dieser repräsentiert die maximale, zwischen zwei Komponenten bestehende Temperaturdifferenz im gesamten NoC und gibt damit Auskunft über die Balance des Temperaturprofils. Ein reduzierter Wert für $Akt_{Schwell}$ (d. h. K2.1) liefert auch diesbezüglich das beste Resultat und würde die Differenz im Vergleich zur Referenz um 5°C und im Vergleich zum besten reaktiven Resultat (d. h. K1.3) um $4,7^{\circ}\text{C}$ reduzieren. Dies entspräche einer Erhöhung der Lebensdauer um 41 % beziehungsweise 38 %.

Vergleich der besten Konfigurationen Ein direkter Vergleich zwischen den besten Konfigurationen für das proaktive und das reaktive Temperaturmanagement bezüglich der Maximaltemperatur ist in Abbildung 4.9 illustriert. Wie erkennbar ist, folgt die Kurve der Maximaltemperatur für das reaktive Management annähernd der Referenzkurve. Dies heißt nicht, dass der reaktive Ansatz generell nicht in der Lage ist Temperaturreduzierungen zu bewirken, da sich der Wert der Maximaltemperatur auf das gesamte System bezieht. Aus der Perspektive einzelner Systemkomponenten ergeben sich in jedem Fall zeitweise Temperaturreduzierungen. Die Maximaltemperatur des Systems mit proaktivem Temperaturmanagement liegt hingegen mit einem Abstand von durchschnittlich $6,7^{\circ}\text{C}$ deutlich unterhalb dieser beiden Kurven.

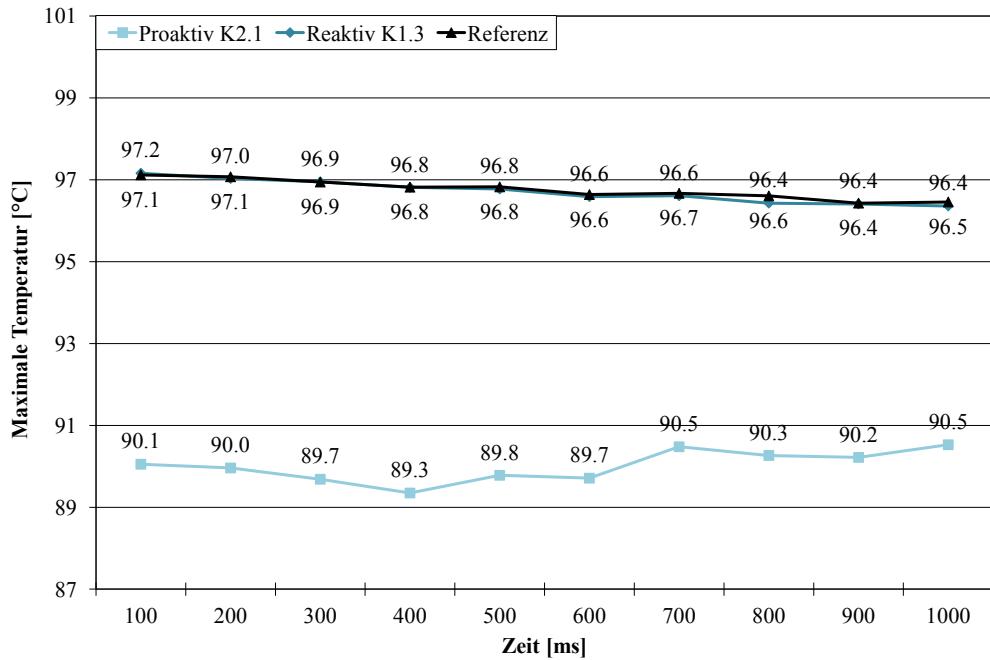


Abbildung 4.9: Verlauf der Maximaltemperatur in einem 4×4 NoC über die Dauer einer Sekunde unter Verwendung der jeweils günstigsten Konfiguration für proaktives und reaktives Management

Aus Abbildung 4.9 geht hervor, dass das proaktive Temperaturmanagement bezüglich einer dauerhaften Temperaturreduzierung wie gewünscht arbeitet. Dies bedeutet, dass nicht nur die über die gesamte Simulationsdauer ermittelten Durchschnittswerte unter denen der Referenz und des reaktiven Ansatzes liegen. In diesem Fall kann darüber hinaus beispielsweise auch die Maximaltemperatur dauerhaft niedriger gehalten werden. Damit werden nicht nur Temperaturspitzen reduziert, es wird auch thermischer Stress durch Temperaturschwankungen vermieden. Dieser thermische Stress würde entstehen, falls das proaktive Management nur kurzzeitige Temperaturreduzierungen bewirken würde und die Temperatur dann wieder auf das Ausgangsniveau zurückkehren würde. Im Mittel würde bei Betrachtung der Durchschnittswerte trotz dessen der Eindruck eines effektiven Temperaturmanagements entstehen.

Das proaktive Temperaturmanagement profitiert am stärksten von der Reduzierung von $Akt_{Schwell}$, während für das reaktive Management hohe Werte für $T_{Schwell}$ die besten Ergebnisse erzeugen. Zudem sollte $T_{Schwell}$ für ein proaktives Temperaturmanagement verhältnismäßig klein gewählt werden. Dies gilt auch für $T_{S,TMU}$, da die Aktualität des prognostizierten Temperaturverlaufs in nicht zu vernachlässigender Weise Einfluss auf die Resultate hat. Ob diesbezüglich Kompromisse zugunsten der Erhaltung einer hohen Leistungsfähigkeit des Systems notwendig sind, wird im weiteren Verlauf untersucht.

Dies ist notwendig, da auch die Überwachungsnachrichten und Steuerungsinstruktionen zur Erwärmung beitragen und darüber hinaus zusätzlich das Netzwerk belasten.

Zeitliche Aspekte

Neben dem eigentlichen Ziel der Reduzierung der Temperatur NoC-basierter Systeme dürfen auch andere Faktoren nicht vernachlässigt werden. Dazu zählen auch zeitliche Aspekte, welche durch die Strategie des Temperaturmanagements maßgeblich beeinflusst werden.

Dies betrifft sowohl die Dauer der Paketübertragung D_P als auch die Router-Verzögerung D_R . Ersteres gibt die Dauer an, die benötigt wird, um ein Paket vollständig von der Quelle zum Ziel zu übertragen. Die Zeitmessung beginnt mit der Einspeisung des Header-Flits in das Netzwerk und endet mit dem Eintreffen des letzten Flits am Ziel. Letzteres beschreibt die Verzögerung der Weiterleitung eines einzelnen Flits durch einen Router, welche unter anderem durch die Arbitrierung und das Switching entsteht. Weiterhin wird die Dauer der durch DFS reduzierten Taktfrequenz der Router und IPCs betrachtet.

Dauer der Paketübertragung Die auf die Referenz normierten Resultate für D_P sind in Abbildung 4.10 dargestellt. Die Latenzen erhöhen sich für beide Ansätze aufgrund der durchgeführten Überwachung und Steuerung deutlich, sodass D_P in jedem Fall mindestens verdoppelt wird. Eine Paketübertragung nimmt damit durchschnittlich die doppelte Zeit in Anspruch.

Besonders schlecht schneidet dabei das reaktive Management für den Fall niedriger Werte für $T_{Schwell}$ ab. Dies ist einerseits darauf zurückzuführen, dass niedrige Werte für $T_{Schwell}$ ein erhöhtes Aufkommen von Überwachungsnachrichten verursachen. Diese erhöhen die Belastung des Netzwerks und begünstigen die Temperaturerhöhung. Zudem werden unzulässige Temperaturwerte häufiger der TMU gemeldet, sodass mehr Steuerungsinstruktionen generiert werden. Diese wiederum bewirken eine Reduzierung der Leistungsfähigkeit. Andererseits sorgt die erhöhte Auslastung des Netzwerks dafür, dass beispielsweise Steuerungsinstruktionen verzögert am Ziel eintreffen. Diese Instruktionen dienen unter anderem der Herstellung des Normalbetriebs einzelner Komponenten, nachdem eine Abkühlung zu verzeichnen ist. Eine Verzögerung bedeutet daher eine unnötig lange Dauer eingeschränkter Leistungsfähigkeit.

Für das proaktive Temperaturmanagement erhöht sich hingegen mit steigenden Werten für $T_{Schwell}$ die Dauer der Paketübertragung. Dieses zum reaktiven Ansatz inverse Verhalten hat seinen Ursprung darin, dass $T_{Schwell}$ anstatt der Anzahl der Überwachungsnachrichten die Anzahl der Steuerungsinstruktionen beeinflusst. Durch höhere Werte für $T_{Schwell}$ werden somit nicht nur temperaturreduzierende Maßnahmen, son-

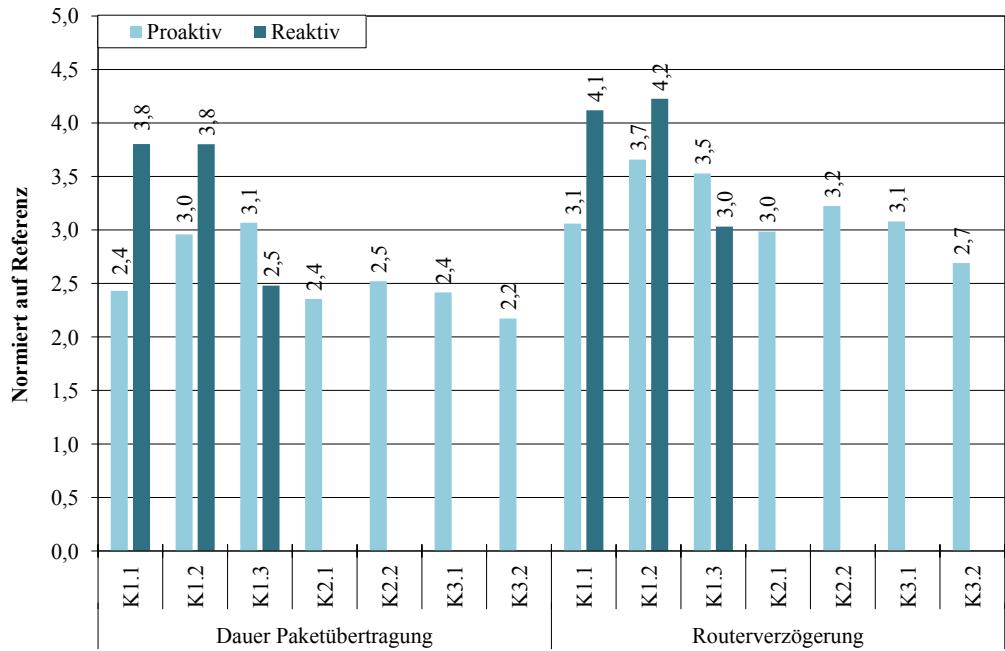


Abbildung 4.10: Durchschnittswerte für die Dauer einer Paketübertragung (von Start zu Ziel) und die Verzögerung der Weiterleitung eines Flits durch einen Router in einem 4×4 NoC unter Verwendung verschiedener Konfigurationen (normiert auf Referenz ohne Management, höhere Werte bedeuten eine längere Zeitspanne)

dern auch leistungswiederherstellende Maßnahmen behindert. Zudem besteht die Gefahr, dass den durch einen höheren Schwellwert prognostizierten Temperaturanstiegen nicht im Vorfeld entgegen gewirkt werden kann und eine Reduzierung der Leistungsfähigkeit tatsächlich länger notwendig ist.

Eine Änderung von $Akt_{Schwell}$ hat trotz deutlicher Unterschiede bezüglich des generierten Verkehrsvolumens für die Überwachung (siehe Tabelle 4.3) nur geringfügige Auswirkungen. Dies deutet darauf hin, dass die Steuerungsinstruktionen das zeitliche Verhalten des Systems am signifikantesten beeinflussen. Diese Vermutung wird dadurch untermauert, dass D_P selbst bei einer stark reduzierten Anzahl von Überwachungsnachrichten (siehe K2.2) nicht reduziert werden kann.

Eine Anpassung von $T_{S, TMU}$ hat ebenfalls kaum Einfluss auf D_P . Da $T_{S, TMU}$ die Aktualisierungshäufigkeit des TMU-internen Temperaturprofils steuert, ist dies auf eine reduzierte Anzahl als unzulässig detekter Temperaturänderungen zurückzuführen. Mit deren Anzahl reduziert sich auch die Anzahl der generierten Instruktionen.

Dauer der Router-Verzögerung Grundsätzlich treffen diese Ausführungen auch auf die Router-Verzögerung D_R zu. Für diese wird jedoch mindestens die 2,7-fache durchschnittliche Dauer für die Weiterleitung eines einzelnen Flits durch einen Router benötigt

(siehe Abbildung 4.10). Für das reaktive Management erhöhen sich die Durchschnittswerte auf mehr als das Vierfache. Allein auf D_R (und auch D_P) bezogen liefert die Konfiguration K3.2 die besten Resultate.

Inwiefern dies auch mit leistungsbezogenen Systemparametern korreliert, müssen weitere Untersuchungen zeigen. Bezuglich einer Reduzierung des thermischen Stresses haben die vorangegangenen Betrachtungen allerdings bereits gezeigt, dass eine verlängerte Simulationsperiode kontraproduktiv ist.

Dauer reduzierter Taktfrequenz Zusätzlich zu D_P und D_R gibt auch die Dauer der durch DFS reduzierten Taktfrequenz einzelner Systemkomponenten Aufschluss über die Effizienz der Managementstrategien in Verbindung mit den verschiedenen Konfigurationen. Die dynamische Taktanpassung ist derart gestaltet, dass die Frequenz in 10 %-Schritten der regulären Betriebsfrequenz zwischen dieser und der halben Betriebsfrequenz gewählt werden kann. Bei einer Betriebsfrequenz von beispielsweise 500 MHz ergibt dies 50 MHz-Schritte innerhalb einer Spanne von 250 bis 500 MHz.

In Abbildung 4.11 ist die Dauer der reduzierten Taktfrequenz sowohl für Router als auch für IPCs dargestellt. Die insgesamt betrachtete Simulationsdauer beläuft sich auf eine Sekunde. Auffällig ist, dass im Rahmen des reaktiven Managements die Router im Durchschnitt fast dauerhaft mit reduzierter Frequenz arbeiten. Dieses Verhalten zeigt sich für alle gewählten Werte von T_{Schw} und weist einerseits auf generell zu scharfe Bedingungen für eine Drosselung der Betriebsfrequenz hin, sodass die Router unnötig oft und übermäßig lange mit reduzierter Frequenz arbeiten. Andererseits sind derartige Ergebnisse die logische Konsequenz aus dem Unvermögen des reaktiven Managements die Temperatur zu reduzieren und das Temperaturprofil des Systems auszubalancieren (siehe Abbildungen 4.8 und 4.9).

Einen ähnlichen Effekt haben steigende Werte für T_{Schw} auf das proaktive Management. Bei zu großen Werten (siehe K1.3) wird kaum eine Rückkehr in den Normalbetrieb veranlasst. Am wirkungsvollsten kann die Dauer der reduzierten Betriebsfrequenz der Router durch niedrige Werte für Akt_{Schw} minimiert werden, sodass die Router durchschnittlich mehr als die Hälfte der Zeit mit voller Leistungsfähigkeit arbeiten. Eine Anpassung von $T_{S,TMU}$ hat ebenfalls starke Auswirkungen auf die Dauer der Reduzierung. Kurze Perioden begünstigen die Detektion von Temperatursenkungen. Lange Perioden haben hingegen genau das Gegenteil zur Folge, sodass die TMU über längere Zeiträume keine Wiederherstellung der vollen Leistungsfähigkeit veranlasst.

Die bezüglich der Router identifizierten Tendenzen treffen prinzipiell auch auf die Dauer der reduzierten Taktfrequenz der IPCs zu. Ein Unterschied besteht darin, dass das reaktive Management für erhöhte Werte von T_{Schw} keinerlei Leistungsreduzierung veranlasst. Dies deutet darauf hin, dass für die IPCs die höheren Schwellwerte nicht über-

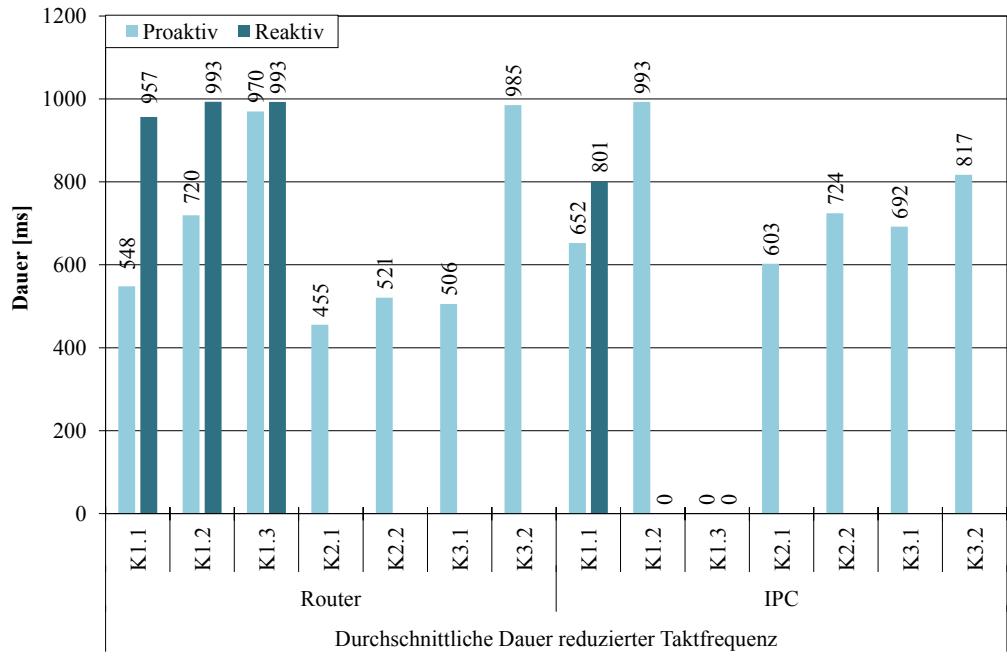


Abbildung 4.11: Durchschnittliche Dauer der durch DFS reduzierten Taktfrequenz der IPCs und Router in einem 4×4 NoC für das proaktive und das reaktive Temperaturmanagement unter Verwendung verschiedener Konfigurationen

schritten werden und somit auch keine Maßnahmen ergriffen werden, welche ansonsten aus entsprechenden Meldungen der Sonden an die TMU resultieren würden.

Dies trifft auch auf das proaktive Management zu, wobei hier nur der höchste Schwellwert (siehe K1.3) diesen Effekt erzielt. Für diesen kann allerdings die Temperatur in keiner Weise positiv beeinflusst werden (siehe Abbildung 4.8). Die Konfiguration K1.2 hingegen hat für die gesamte Simulationsdauer eine reduzierte Leistungsfähigkeit zur Folge. Der Schwellwert wird für K1.2 zwar überschritten, aber die Maßnahmen zur Wiederherstellung der vollen Leistungsfähigkeit werden aufgrund konstant hoher Temperaturwerte nicht eingeleitet (siehe auch Abbildung 4.8). Auch für die IPCs kann die Dauer der Frequenzdrosselung am effizientesten mit niedrigen Werten für $Akt_{Schwell}$ reduziert werden. Diese arbeiten trotz dessen während 60 % der Gesamtdauer mit reduzierter Frequenz. Eine Änderung von $T_{S,TMU}$ bewirkt auch für die IPCs eine Verschlechterung.

Aus der Sicht einer möglichst hohen Leistungsfähigkeit betrachtet, ist demnach allgemein die Konfiguration eines temperaturorientierten Systemmanagements ausschlaggebend dafür, ob sich der Einsatz eines solchen Managements lohnt. Andernfalls ist in Betracht zu ziehen, die Leistungsfähigkeit des Gesamtsystems dauerhaft zu senken, um auf diese Weise gegebenenfalls ähnliche Temperaturreduzierungen bei geringeren Leistungseinbußen zu erzielen.

Die durchgeführten Untersuchungen zeigen, dass sowohl durch das reaktive als auch das proaktive Temperaturmanagement teils erhebliche Verzögerungen in Kauf genommen werden müssen. Durch die gezielte Anpassung einzelner Parameter ist es aber möglich diese Verzögerungen zumindest für den proaktiven Ansatz zu reduzieren.

In diesem Zusammenhang erweisen sich niedrige Werte für $Akt_{Schwell}$ als am besten geeignet. Falls die Leistungsfähigkeit der Router ein entscheidendes Kriterium darstellt, sind zudem kleine Werte für $T_{Schwell}$ empfehlenswert. Andernfalls treten starke Verzögerungen auf. Für $T_{S, TMU}$ sind in diesem Fall dagegen große Werte zu bevorzugen, wobei diese sich allerdings negativ auf die Temperatur auswirken.

Die reaktive Managementstrategie weist trotz vergleichsweise schlechter Resultate bezüglich der Reduzierung des thermischen Stresses unerwartet auch an dieser Stelle signifikante Defizite auf. Inwiefern dies auch für die Leistungsfähigkeit des zugrunde liegenden Systems gilt, wird nachfolgend untersucht.

Leistungsfähigkeit des Systems

Im Zuge der durchgeführten Untersuchungen spielt auch die eigentliche Leistungsfähigkeit des Systems in Form der durchschnittlichen Netto-Datenrate eine grundlegende Rolle. Diese ergibt sich aus der Anzahl der in einem Zeitabschnitt erfolgreich zum Ziel übertragenen Bits abzüglich der in den Flits enthaltenen Steuerinformationen (z. B. Zieladresse oder Paketlänge). Dieser Wert wird durch die Dauer des Zeitabschnitts dividiert. Die Überwachungs- und Steuerungsnachrichten werden ausgeschlossen, um Verfälschungen der Resultate zu vermeiden.

Die Werte der Datenrate geben Aufschluss darüber, wie stark das Temperaturmanagement den regulären Betrieb durch zusätzlichen Verkehr und eventuell durchgeführte Maßnahmen zur Temperaturreduzierung behindert. Da eine Beeinträchtigung grundsätzlich unvermeidlich ist, gilt es die Auswirkungen des proaktiven und reaktiven Temperaturmanagements genauer zu untersuchen. Auf diese Weise lassen sich Konfigurationen identifizieren, welche die Behinderung auf ein Minimum reduzieren und dennoch annehmbare Temperaturminderungen bewirken. Die Ergebnisse bezüglich der Netto-Datenrate unter Verwendung verschiedener Konfigurationen sind in Abbildung 4.12 dargestellt und wurden auf ein Referenzsystem ohne Management normiert. Die Injektionsrate eines einzelnen IPC wurde auf 20 % festgelegt. Dies ist von Bedeutung, da diese Rate die Häufigkeit bestimmt, mit welcher ein IPC Daten in das Netzwerk einspeist und somit die Netto-Datenrate beeinflusst. Neben der Datenrate bietet sich für eine Analyse der Leistungsfähigkeit auch die Betrachtung der Gesamtanzahl der übertragenen Pakete an. Die auf das Referenzsystem normierten Ergebnisse sind allerdings mit den Resultaten für die Datenrate identisch und werden daher nicht gesondert behandelt.

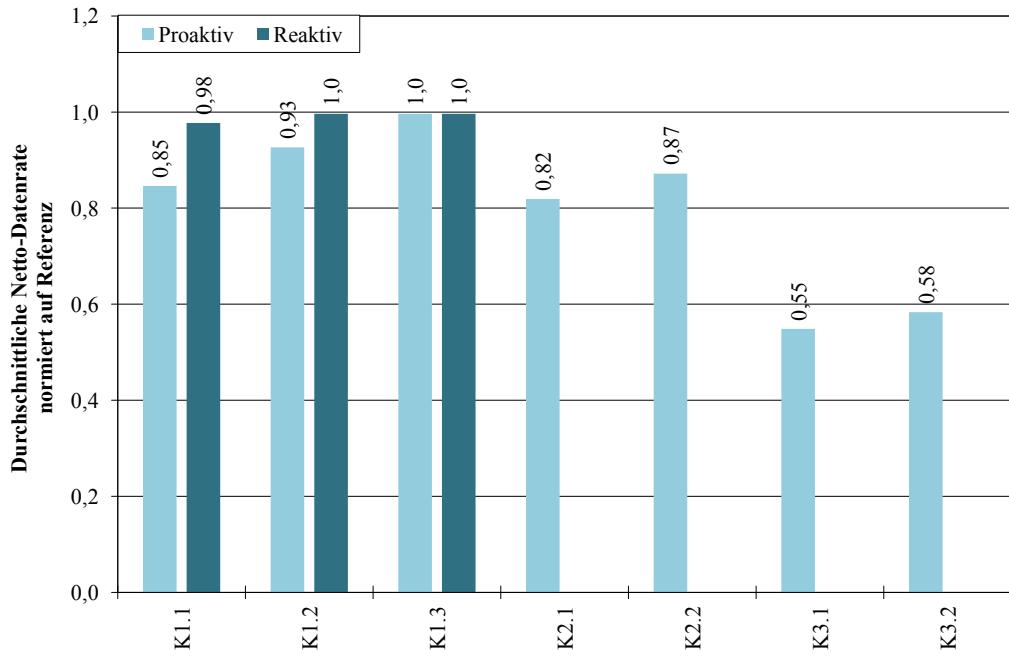


Abbildung 4.12: Durchschnittliche Netto-Datenrate in einem 4×4 NoC unter Verwendung verschiedener Konfigurationen (normiert auf Referenz ohne Management, Injektionsrate je IPC: 20 %)

Bezüglich der Datenrate wird deutlich, dass das reaktive Management in der Lage ist Einbußen fast vollständig zu vermeiden. Im Hinblick auf das Unvermögen Temperaturreduzierungen zu erzielen (siehe Abbildung 4.8) und den vergleichsweise geringen zusätzlichen Verkehr durch Überwachung und Steuerung (siehe Tabelle 4.3) ist dies die logische Konsequenz. Mit steigenden Werten für $T_{Schwell}$ erreicht auch das proaktive Management den Normalwert. Die Einbußen durch den kleinsten überprüften Schwellwert sind auf diejenigen IPCs zurückzuführen (siehe Abbildung 4.11), welche durch Frequenzanpassungen verlangsamt arbeiten. Dadurch wird zwar weniger Datenverkehr verarbeitet, die Temperatur kann allerdings nur auf diese Weise nennenswert reduziert werden (siehe Abbildung 4.8). Auch die Anzahl der Steuerungsinstruktionen nimmt mit kleineren Werten für $T_{Schwell}$ zu. Diese Zahl ist im Vergleich zu den Überwachungsnachrichten allerdings verschwindend gering.

Die Anpassung des Schwellwertes $Akt_{Schwell}$ ermöglicht eine Anhebung der Datenrate auf bis zu 87 % des Ausgangswertes. Kleine Schwellwerte reduzieren dabei die erzielbare Datenrate, während große Werte diese erhöhen. Die Ursache hierfür liegt in der Anzahl der generierten Überwachungsnachrichten, welche mit kleineren Schwellwerten stark zunimmt. Auf diese Weise wird das Netz zusätzlich belastet, die Übertragung regulärer Daten wird erschwert und die Datenrate sinkt. Falls also die Leistungsfähigkeit des Systems im Vordergrund steht, würde eine Anhebung von $Akt_{Schwell}$ zielführend sein.

Wird der Schwellwert jedoch zu hoch gewählt, geht möglicherweise der zeitliche Vorteil des proaktiven Managements verloren (siehe Abschnitt 4.2.1), da detektierte Aktivitäten unter Umständen lange zurückgehalten werden, bevor es zu einer Meldung kommt.

Eine Modifikation der TMU-internen Simulationsperiode $T_{S, TMU}$ hat in jedem Fall drastische Reduzierungen der Leistungsfähigkeit zur Folge, sodass die Netto-Datenrate auf weniger als 60 % des Ausgangswertes sinkt. Dafür verantwortlich ist die lange Dauer, die IPCs und Router durchschnittlich im Modus reduzierter Leistungsfähigkeit verbringen. Diese Dauer wird durch hohe Werte für $T_{S, TMU}$ verlängert, da die Frequenz der Aktualisierung des TMU-internen Temperaturprofils herabgesetzt wird. Somit können Maßnahmen zur Wiederherstellung der Leistungsfähigkeit im Falle des Absinkens der Temperatur nur verspätet durchgeführt werden. Die Router verlangsamen durch eine reduzierte Betriebsfrequenz die Weiterleitung der Daten im Netz und die IPCs versenden und empfangen Daten langsamer.

Am effizientesten lässt sich der Nachteil eines verringerten Datendurchsatzes für das proaktive Management durch eine Anhebung von $T_{Schwell}$ kompensieren. Dadurch vermindert sich allerdings die Temperaturreduktion deutlich, sodass kleine Schwellwerte attraktiver erscheinen. Zudem sind Einbußen von etwa 15 % für kleine Schwellwerte im Hinblick auf die Verlängerung der TTF um etwa 20 % akzeptabel. Ein hoher Schwellwert für die Generierung von Überwachungsnachrichten trägt ebenfalls deutlich zu einer Reduzierung des Leistungsverlustes bei. Auch hier gilt, dass die durch hohe Schwellwerte erzielte Wiederherstellung der Datenrate durch eine Erhöhung des thermischen Stresses erkauft wird. Je nachdem welcher Faktor von primärer Bedeutung ist, muss also ein Kompromiss zwischen Leistung und Temperaturreduktion eingegangen werden.

Managementaufwand

In den durchgeführten Untersuchungen ist bereits deutlich geworden, dass auch die Intensität, mit der ein Temperaturmanagement betrieben wird, einen nicht zu vernachlässigenden Einfluss auf die Gesamtfunktionalität eines Systems sowie dessen Temperaturentwicklung hat. Je nach Anzahl der Überwachungs- und Steuerungsnachrichten können verschiedene Effekte eintreten.

Eine hohe Anzahl von Überwachungsnachrichten kann bis zu einem gewissen Grad die Genauigkeit der Temperaturüberwachung begünstigen und auf diese Weise zu einer effizienteren Steuerung beitragen. Dies ist auf aktuellere und genauere Informationen zurückzuführen, über welche die TMU verfügen kann. Allerdings hat eine hohe Anzahl von Überwachungsnachrichten auch eine erhöhte Auslastung des Netzwerks zur Folge. Dies führt neben einer Behinderung des regulären Verkehrs auch zu einer weiteren Erwärmung durch zusätzliche Schaltaktivitäten. Zudem werden möglicherweise sich be-

reits im Netzwerk befindliche Steuerungsinstruktionen blockiert, welche dadurch unter Umständen erst verspätet eintreffen. Werden hingegen zu wenige Überwachungsnachrichten generiert, besteht die Gefahr, dass die TMU mit veralteten Informationen versorgt wird und falsche Entscheidungen trifft. Das Netzwerk wird dabei entlastet.

Trotz des im Vergleich zum Datenvolumen (siehe Tabelle 4.3) geringen Anteils der Überwachungsnachrichten und Steuerungsinstruktionen ist ein Auftreten der beschriebenen Mechanismen besonders dann wahrscheinlich, wenn durch starke temporäre Schaltaktivitäten stoßartig viele Überwachungsnachrichten gleichzeitig erzeugt werden. Detaillierte Resultate bezüglich der Anzahl der Überwachungs- und Steuerungsnachrichten sowie des jeweiligen Datenvolumens sind in Tabelle 4.3 aufgeführt.

Wie erkennbar ist, liegt der Umfang des im Rahmen des proaktiven Managements generierten Überwachungsverkehrs um mehrere Potenzen höher als der des reaktiven Gegenstücks. Dies ist darauf zurückzuführen, dass anstelle von sich träge ändernden Temperaturen Aktivitätswerte überwacht werden. Diese treten naturgemäß mit weitaus höherer Frequenz auf, da nicht jede Schaltaktivität auch einen sofortigen Temperaturanstieg bewirkt. Dass die im Vergleich zum reaktiven Management hohe Zahl der Überwachungsnachrichten die Temperatur nicht negativ beeinflusst, haben bereits vorausgegangene Untersuchungen gezeigt. Bemerkenswert ist jedoch, dass sowohl die Standardeinstellungen als auch reduzierte und erhöhte Aktivitätsschwellwerte (K1.1, K2.1 und K2.2) trotz stark unterschiedlicher Ergebnisse bezüglich der Anzahl der Überwachungsnachrichten eine fast identische Anzahl generierter Steuerungsinstruktionen aufweisen. Dies deutet darauf hin, dass ein Teil der Überwachungsnachrichten möglicherweise überflüssig ist und entfallen kann. Eine solche Maßnahme würde das Netzwerk entlasten und Kapazitäten für den regulären Datenverkehr freigeben. Auf thermische Aspekte hat dies kaum einen Einfluss, wohl aber auf die Menge der übertragenen regulären Daten.

Die angestellten Betrachtungen haben gezeigt, dass der für ein Temperaturmanagement betriebene Aufwand sowohl die Effizienz dieses Managements als auch die Leistungsfähigkeit des zugrunde liegenden Systems beeinflusst. Im Einzelnen bedeutet dies, dass ein Kompromiss zwischen der Sensitivität einer Überwachung und der daraus resultierenden Behinderung des regulären Datenverkehrs getroffen werden muss.

Steht die Temperaturreduzierung im Vordergrund, müssen daher die Überwachungsmechanismen empfindlicher auf Veränderungen reagieren. Nur auf diese Weise kann die TMU schnellstmöglich benachrichtigt werden, sodass entsprechende Maßnahmen getroffen werden können. Dabei ist allerdings darauf zu achten, eine Übersensibilisierung und damit einhergehenden überflüssigen Überwachungsverkehr zu vermeiden. Dies ist besonders für das proaktive Management von Bedeutung, da aufgrund der Aktivitätsüberwachung wesentlich häufiger entsprechende Nachrichten erzeugt werden.

Tabelle 4.3: Ergebnisse für die Anzahl der Überwachungsnachrichten, der Steuerungsstrukturen und der übertragenen regulären Datenpakete in einem 4×4 NoC unter Verwendung verschiedener Konfigurationen

		K1.1	K1.2	K1.3	K2.1	K2.2	K3.1	K3.2
Überwachung	Proaktiv	185 413	207 991	223 235	364 624	93 840	105 682	112 183
	Reaktiv	705	396	253	/	/	/	/
Steuerung	Proaktiv	356	311	255	359	357	242	186
	Reaktiv	375	307	247	/	/	/	/
Datenpakete [Mio.]	Proaktiv	171,3	187,5	201,7	165,7	176,5	171,5	182,4
	Reaktiv	197,8	201,7	201,7	/	/	/	/

Ist hingegen die Leistungsfähigkeit des Systems von primärem Interesse, so gilt es die Sensitivität der Überwachung zu verringern. Der auf diese Weise reduzierte Überwachungsverkehr trägt zur Entlastung der Kommunikationsstrukturen bei, sodass der reguläre Betrieb ungehinderter ablaufen kann.

4.2.3 Zusammenfassung

Die durchgeführten Untersuchungen haben gezeigt, dass unabhängig von der eingesetzten Strategie für das Temperaturmanagement NoC-basierter Systeme immer ein Kompromiss zwischen der erzielten Temperaturreduzierung und den damit einhergehenden Einbußen bezüglich der Leistungsfähigkeit eingegangen werden muss. Je nachdem welcher Faktor höhere Priorität genießt, sind mithilfe ausgewählter Managementparameter dementsprechende Anpassungen durchführbar. Auf diese Weise lässt sich das Temperaturmanagement auf die Anforderungen des zugrunde liegenden Systems abstimmen.

Obwohl in dieser Arbeit nicht näher darauf eingegangen wird, ist diesbezüglich auch eine dynamische Anpassung denkbar. Auf diese Weise könnten hoch priorisierte, beispielsweise mit Echtzeitanforderungen behaftete, Aufgaben bevorzugt abgearbeitet werden. Weniger kritische Aufgaben würden zugunsten einer reduzierten thermischen Belastung gezielt zurückgestellt. Die proaktive Strategie bietet dabei im Vergleich zum reaktiven Ansatz sowohl für ein statisches als auch ein dynamisch anpassbares Temperaturmanagement deutlich mehr Spielraum. Dies gilt zumindest für die durchgeführten Betrachtungen und untersuchten Variationen der Parameter des Managements.

Im Rahmen dieser Betrachtungen konnte gezeigt werden, dass das proaktive Management einerseits durch entsprechende Konfigurationen in der Lage ist, die durch ein Temperaturmanagement verursachten Leistungsdefizite zu reduzieren (siehe Abbildung

Tabelle 4.4: Durchschnittliche Temperatur in NoCs verschiedener Größe unter Verwendung der Standardkonfiguration K1.1 für das proaktive sowie das reaktive Temperaturmanagement und ein äquivalentes System ohne Management

K1.1	2×2	3×3	4×4	5×5	6×6
Reaktiv	58,3	65,4	71,5	74,0	74,2
Proaktiv	58,2	64,2	69,3	72,4	75,0
Referenz	58,5	65,5	71,9	76,4	80,6

gen 4.10 und 4.12). Dabei kann der positive Effekt auf die Temperaturverteilung zumindest teilweise aufrecht erhalten werden.

Andererseits ist es ebenfalls möglich den thermischen Stress wesentlich stärker zu reduzieren. Im Gegenzug müssen allerdings auch stärkere Leistungseinbußen in Kauf genommen werden. Die reaktive Strategie schneidet in dieser Hinsicht deutlich schlechter ab und kann kaum eine Senkung der Temperatur bewirken. Die Ursache hierfür liegt in den beschränkten Variationsmöglichkeiten der Managementparameter (lediglich $T_{Schwell}$ ist modifizierbar), welche sich aus dem Fokus der Untersuchungen ergeben. Dieser richtet sich nicht auf die Umsetzung der eigentlichen Steuerungsmaßnahmen, sondern auf die Effizienz der Kommunikation zwischen Überwachungssystem und TMU sowie auf die Ermittlung unzulässiger Temperaturwerte. Diese Faktoren sind der markanteste Unterschied zwischen dem vorgestellten proaktiven Ansatz und herkömmlichen reaktiven Strategien. Unter Umständen begünstigen beispielsweise Parameter zur Steuerung der TMU-internen Funktionalität (z. B. Auslösen der Verlagerung von Aufgaben) das reaktive Management stärker. Zudem beschränken sich die Resultate auf ein 4×4 NoC.

Dass sich die Verhältnisse auch mit sich ändernden NoC-Dimensionen verschieben können, belegen die Ergebnisse in Tabelle 4.4. In dieser ist die Durchschnittstemperatur in Systemen verschiedener Größe für das proaktive und das reaktive Management unter Verwendung der Konfiguration K1.1 sowie für ein Referenzsystem ohne Management aufgeführt. Während beide Ansätze für ein 2×2 NoC kaum Verbesserungen erwirken, erzielt das proaktive Management für ein 5×5 NoC die deutlich besseren Ergebnisse. Für ein 6×6 NoC ist hingegen der reaktive Ansatz im Vorteil. Dieses Verhalten ist auf das komplexe Zusammenspiel der Managementparameter sowie eine enge Verknüpfung von Leistungsfähigkeit und thermischem Verhalten und daraus resultierenden Wechselwirkungen zurückzuführen.

Unter der Voraussetzung, dass ein möglichst leistungsfähiges System gewünscht ist, sind sowohl für die proaktive als auch die reaktive Managementstrategie hohe Schwellwerte für das Erkennen unzulässiger Temperaturwerte erforderlich. Die proaktive Stra-

Tabelle 4.5: Qualitative Bewertung des Einflusses des proaktiven Temperaturmanagements auf Temperatur und Leistungsfähigkeit für den Fall der Priorisierung einer Temperaturreduzierung

Management	T_{\emptyset}	T_{Max}	ΔT_{Max}	Leistung
Kein	$T_{\emptyset,K}$	$T_{Max,K}$	$\Delta T_{Max,K}$	X
Reaktiv	$T_{\emptyset,K} - \Delta T_R + \epsilon T_R$	$T_{Max,K}$	$\Delta T_{Max,K}$	$X - \Delta X_R$
Proaktiv	$T_{\emptyset,K} - (n \cdot \Delta T_R) + \epsilon T_P$	$T_{Max,K} - n \cdot \Delta T_P$	$\Delta T_{Max,K} - n \cdot \Delta T_P$	$X - \Delta X_R - f(n, \Delta T_P)$

tegie profitiert diesbezüglich ebenfalls von hohen Grenzwerten für das Auslösen temperaturreduzierender Maßnahmen. Eine Verringerung der Aktualisierungsfrequenz des TMU-internen Temperaturprofils wirkt hingegen kontraproduktiv. Steht hingegen eine Temperaturoptimierung im Vordergrund, sind für beide Strategien naturgemäß niedrige Schwellwerte für das Erkennen unzulässiger Temperaturen erforderlich.

In Tabelle 4.5 ist eine qualitative Bewertung des proaktiven Managements im Vergleich zu einem reaktiven Konzept sowie einem System ohne jegliches Management für den Fall aufgeführt, dass eine Reduzierung des thermischen Stresses priorisiert wird.

Es wird angenommen, dass die durchschnittliche Temperatur $T_{\emptyset,K}$ durch eine reaktive Strategie um einen Wert ΔT_R reduziert werden kann. Die durch das reaktive Management entstehende Abwärme wird durch ϵT_R repräsentiert und schlägt sich in der Bilanz für $T_{\emptyset,K}$ entsprechend nieder. Die durchgeführten Untersuchungen konnten zeigen, dass die proaktive Strategie in der Lage ist ΔT_R um einen Faktor n zu steigern. Dieser Faktor hängt von einer Reihe von Parametern ab und ermöglicht je nach Konfiguration entweder eine Temperaturreduzierung oder eine Verminderung der Leistungseinbußen. Die zusätzlich erzeugte Abwärme ϵT_P muss dabei entsprechend gering gehalten werden.

Die reaktive Strategie ist nicht beziehungsweise kaum in der Lage, die maximale Temperatur T_{Max} sowie den maximalen Temperaturunterschied ΔT_{Max} zu reduzieren. Das proaktive Konzept erreicht hingegen für beide Parameter eine Verbesserung von ΔT_P , welche etwa in gleichem Maße wie T_{\emptyset} durch n beeinflusst wird.

Die Kehrseite des reduzierten thermischen Stresses ist eine durch beide Strategien verminderte Leistungsfähigkeit. Das proaktive Management weist in diesem Zusammenhang jedoch deutlich mehr Spielraum auf. Damit lässt sich die bereits angesprochene Priorisierung zwischen reduziertem thermischem Stress und verminderter Leistungseinbußen realisieren, da auch diese Einbußen von n sowie ΔT_P abhängen.

4.3 Anwendungsszenario II: Proaktives Task-Mapping

Mehr kernprozessoren bieten aufgrund der Integration einer wachsenden Anzahl von IPCs auf einem einzelnen Chip die Möglichkeit mehrere Anwendungen parallel zu bearbeiten. Je nach Auslegung der Systemkomponenten und den Leistungsanforderungen der Anwendungen konkurriert dabei eine unterschiedlich große Anzahl von Prozessen um die verfügbaren Systemressourcen. Zudem unterteilen sich die Anwendungen oftmals in viele kleinere Teilaufgaben, sogenannte Tasks, welche untereinander kommunizieren. Für NoC-basierte Mehrkernprozessoren bedeutet dies, dass einerseits durch die Möglichkeit einer parallelen Bearbeitung mehrerer Aufgaben eine hohe Systemleistungsfähigkeit erzielt werden kann. Andererseits steigt damit das Kommunikationsaufkommen durch den Datenaustausch zwischen den auf den IPCs verteilten Tasks.

Die Folgen sind Temperaturungleichgewichte und erhöhte Temperaturen. Solche Effekte können bis zu einem gewissen Grad mithilfe einer geeigneten Mapping-Strategie ausgeglichen werden. Ist ein starres Anwendungsszenario (z. B. ausschließlich Video- oder Audiodekodierung) vorgesehen, so bietet sich ein Mapping bereits während der Entwurfsphase des zugrunde liegenden Systems an. Ist hingegen das Anwendungsszenario nicht exakt bekannt oder muss mit dynamisch variierenden Anwendungsprofilen gerechnet werden, so erscheint ein dynamisches Mapping zur Laufzeit des Systems als der geeignetere Ansatz. Der Grund hierfür ist, dass in die Mapping-Entscheidungen aufgrund der unvorhersehbaren Systemdynamik auch die Verfügbarkeit der Systemressourcen sowie der Zustand der Kommunikationsinfrastruktur einfließen müssen.

In diesem Abschnitt wird daher untersucht, inwieweit sich das in Kapitel 3 vorgestellte Temperaturmodell für ein dynamisches proaktives Task-Mapping zur Laufzeit NoC-basierter Mehrkernprozessoren eignet. Die thermischen Auswirkungen zu platzierender Tasks sollen dabei im Voraus abgeschätzt werden, um ein möglichst ausgeglichenes Temperaturprofil zu erzeugen. In Abschnitt 4.3.1 wird das Konzept des proaktiven Task Mappings vorgestellt. Zudem werden drei gängige, zu Vergleichszwecken herangezogene Mapping-Strategien eingeführt. In Abschnitt 4.3.2 werden die Mapping-Strategien bezüglich ihrer Auswirkung auf das Temperaturprofil des Systems sowie ausgewählte leistungsbezogene Parameter verglichen. Abschließend werden die gewonnenen Erkenntnisse zusammengefasst.

4.3.1 Konzept

Das hier vorgestellte Konzept eines proaktiven temperaturorientierten Task-Mappings zur Laufzeit NoC-basierter Mehrkernprozessoren greift auf das in Kapitel 3 vorgestellte Temperaturmodell zurück. Analog zum proaktiven Temperaturmanagement wird auch dieses Konzept in die Simulationsumgebung (siehe Abschnitt 3.1.3) integriert, welche

Task-Graph

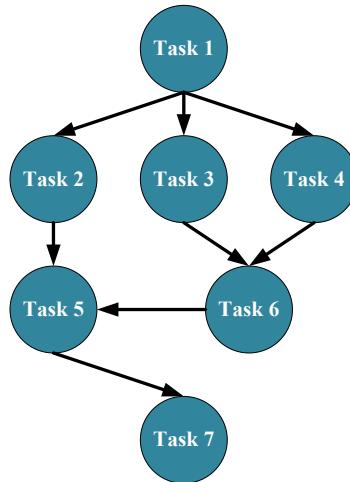


Abbildung 4.13: Anwendungsmodellierung durch einen Task-Graphen: die Abhängigkeit einzelner Tasks von den Ergebnissen anderer Tasks gibt die Reihenfolge der Abarbeitung vor

bereits um das erforderliche Temperaturmodell erweitert wurde. Diese Variante der Simulationsumgebung weist allerdings eine entscheidende Besonderheit auf.

Modellierung von Task-Graphen Die Simulationsumgebung wurde im Rahmen einer studentischen Arbeit [Pas11] am Institut für Angewandte Mikroelektronik und Datentechnik um die Modellierung sogenannter Task-Graphen (TG) erweitert. Solche TGs bestehen aus einzelnen Tasks und repräsentieren die durch das System bearbeiteten Anwendungen. Die Erweiterung basiert auf dem frei verfügbaren Modell der Task-Graphs-For-Free (TGFF) [DRW98]. Dieses Modell ist in der Lage nach gewissen Vorgaben pseudozufällig TGs zu generieren, wodurch die Simulation realitätsnaher Last- und Verkehrs- muster ermöglicht wird. Ein TG wird dabei in mehrere Tasks aufgeteilt. Jeder einzelne Task besitzt bestimmte Eigenschaften. Hierzu zählen Abhängigkeitsverhältnisse zu anderen Tasks, Bearbeitungsdauer und -aufwand und das durch die Kommunikation mit anderen Tasks erzeugte Verkehrsvolumen. Auch das Anfordern von Daten durch den Task aus dem Speicher des Systems wird modelliert. Je nach Systemgröße übernimmt diesbezüglich eine variierende Anzahl von IPCs die Rolle von Speicherelementen.

Ein Beispiel für einen solchen TG ist in Abbildung 4.13 illustriert. Wie erkennbar ist, bedürfen einzelne Tasks zunächst der Fertigstellung vorangegangener Tasks oder sind auf Eingabedaten dieser angewiesen. Somit ergibt sich eine feste Abarbeitungsreihenfolge, aus welcher wiederum eine Mapping-Hierarchie hervorgeht. Erst mit der Beendigung des oder der finalen Tasks gilt der komplette TG als fertiggestellt.

Simulationsumgebung mit proaktivem Mapping-Algorithmus Der generelle Aufbau der Simulationsumgebung sowie die Eingliederung des TGFF-Modells in diese ist in Abbildung 4.14 dargestellt. Die für die Simulation bereitgestellten TGs werden im sogenannten Task-Pool vorgehalten, welcher als FIFO ausgelegt ist. Darin befinden sich alle noch nicht platzierten TGs sowie die TGs, welche eine periodische Abarbeitung erfordern und sich daher wiederkehrend im Task-Pool befinden. Ein Simulationsdurchlauf endet damit entweder nach der Abarbeitung einer vorgegebenen Anzahl von Tasks oder nach einer fest definierten Dauer. Im Task-Pool wartende Tasks werden durch den Mapping-Algorithmus sukzessive auf den IPCs des Systems platziert. Je nach Mapping-Strategie geschieht dies nach unterschiedlichen Kriterien.

Aufgrund der teilweise starken Verflechtungen der Tasks eines TG wird unabhängig von der Strategie immer ein kompletter TG im System platziert, sobald der Algorithmus die Freigabe zum Mapping erhält. Nur auf diese Weise kann garantiert werden, dass es während der Simulation nicht zu Verklemmungen durch dauerhaft wartende Tasks kommt. Sämtliche Mapping-Entscheidungen werden der funktionalen Simulation mitgeteilt, sodass die Tasks ihre Arbeit aufnehmen können.

Die im Laufe der funktionalen Simulation erzeugten Aktivitätsprofile werden analog zum proaktiven Temperaturmanagement alle T_S in das, als „Tatsächliches Profil“ gekennzeichnete, Temperaturmodell eingespeist. Bevor dies jedoch geschieht, übergibt der Mapping-Algorithmus die Mapping-Entscheidungen samt aller relevanten Task-Eigenschaften an das „Vorhersageprofil“. Zu diesen Eigenschaften gehören die Bearbeitungsdauer des Tasks, die Menge der im ausgewählten IPC verarbeiteten Daten, die involvierten IPCs sowie das zu erwartende Kommunikationsvolumen. Die Menge der verarbeiteten Daten ist nötig, um die Schaltaktivität für den betroffenen IPC abzuschätzen. Die Koordinaten der beteiligten IPCs werden zur Bestimmung der Kommunikationspfade und der somit auftretenden Schaltaktivitäten entlang dieser Pfade benötigt. Das zu erwartende Kommunikationsaufkommen wird benötigt, um die Schaltaktivität des zum IPC gehörigen Routers zu ermitteln. Darüber hinaus wird dieser Wert für die Bestimmung der entlang der Kommunikationspfade auftretenden Schaltaktivitäten genutzt.

Ausgehend von diesen Informationen wird die Berechnung der zu erwartenden Auswirkung der Mapping-Entscheidung auf die Temperaturentwicklung bereits eingeleitet, bevor der Task seine Arbeit aufnimmt. Auf diese Weise kann analog zum proaktiven Temperaturmanagement zumindest teilweise auf die Messung der Temperatur durch dedizierte Sensoren verzichtet werden. Eine Kalibrierung und eventuelle Korrekturen müssen aber auch hier mithilfe solcher Temperatursensoren durchgeführt werden.

Die Temperaturentwicklung wird dem Mapping-Algorithmus alle $T_{S,Mapping}$ mitgeteilt, sodass der nächste wartende Task auf dem voraussichtlich kühlstens IPC platziert werden kann. Die Mapping-Entscheidungen werden somit auf Basis des Vorhersagepro-

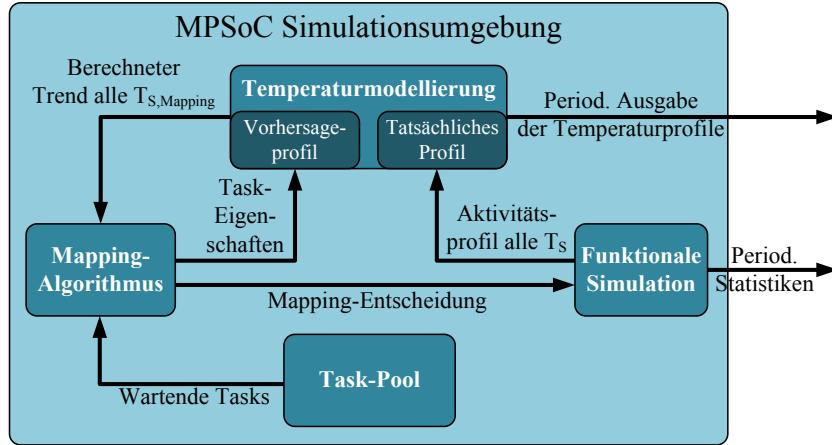


Abbildung 4.14: Umgebung zur Simulation des proaktiven Task-Mappings NoC-basierter Mehrkernprozessoren mithilfe des in Abschnitt 3.1 implementierten Temperaturmodells

falls getroffen. Die Temperaturwerte dieses Profils spielen allerdings nur für den proaktiven temperaturorientierten Mapping-Algorithmus eine Rolle.

Um Aussagen darüber treffen zu können, inwiefern das proaktive temperaturorientierte Task-Mapping in der Lage ist das Temperaturprofil des zugrunde liegenden NoC-basierten Systems positiv zu beeinflussen, werden drei weitere gängige Strategien für einen Vergleich herangezogen. Des Weiteren wird die Auswirkung auf die Leistungsfähigkeit des Systems evaluiert, da durch die Platzierung einzelner, zu einem TG gehöriger Tasks diesbezüglich möglicherweise Einbußen hingenommen werden müssen. Die Ursache hierfür liegt darin, dass eine temperaturoptimierte Mapping-Entscheidung unter Umständen verlängerte Kommunikationspfade zwischen untereinander abhängigen Tasks zur Folge hat. Eine Kommunikation zwischen diesen wird dadurch automatisch verzögert, was ebenfalls die Fertigstellung eines TG hinauszögert.

Vergleichsalgorithmen Neben dem proaktiven Ansatz dient zunächst ein reaktiver temperaturorientierter Algorithmus als Vergleichsstrategie. Dieser nutzt das „Tatsächliche Profil“, um Mapping-Entscheidungen zu treffen. Dabei werden wartende Tasks auf dem tatsächlich kühlstens IPC platziert, sodass das dieses Temperaturprofil möglichst gut ausbalanciert wird. Das Zurückgreifen auf die tatsächlichen Temperaturwerte entspricht dabei der aus Abschnitt 4.2.1 bekannten reaktiven Temperaturmessung mithilfe einzelner Sensoren. Prinzipiell ist dieses Konzept mit dem Mapping in [KCBB06] vergleichbar, nur dass das Mapping dort statisch und bereits während der Entwurfsphase stattfindet.

Des Weiteren kommt ein lastorientierter Algorithmus zum Einsatz, welcher das Ziel hat die Auslastung aller IPCs dynamisch auszubalancieren und dadurch bis zu einem gewissen Grad auch die Temperaturverteilung positiv zu beeinflussen. Zu diesem Zweck

werden wartende Tasks immer auf dem IPC mit den wenigsten bereits zugewiesenen Tasks platziert. Dabei werden allerdings die Eigenschaften der bereits platzierten Tasks nicht berücksichtigt. Unter Umständen wird daher ein IPC mit wenigen, aber dafür aufwändigen und zeitintensiven Tasks einem IPC mit vielen, dafür schnell zu verarbeitenden Tasks vorgezogen. Durch die verteilte Platzierung erhöht sich einerseits die Wahrscheinlichkeit, dass Tasks parallel abgearbeitet werden können. Zudem wird die Temperaturverteilung durch die gleichmäßige Auslastung positiv beeinflusst. Andererseits erhöht sich das Kommunikationsaufkommen im NoC, was sich wiederum negativ auf die Temperatur auswirkt. Dieser Ansatz ist auch als „First-Free“-Algorithmus bekannt und wird beispielsweise in [CRG08] eingesetzt.

Die letzte der Vergleichsstrategien ist ein distanzorientierter Algorithmus. Dieser hat das Ziel, die Kommunikationspfade zwischen untereinander abhängigen Tasks zu minimieren und platziert die Tasks daher möglichst dicht beieinander. Dieses Konzept ist auch als „Nearest-Neighbor“-Algorithmus bekannt und wird beispielsweise in [KSS11] genutzt. Der Nachteil dieser Strategie ist die möglicherweise resultierende Task-Ballung, welche zur Bildung weniger besonders heißer Stellen im System führt. Dies ist besonders dann problematisch, wenn beispielsweise ein TG aufgrund seiner Beschaffenheit seriell abgearbeitet wird. In diesem Fall wartet ein Task in der Regel lediglich auf die Fertigstellung eines vorangegangenen Tasks. Die Folge ist eine Platzierung beider Tasks auf demselben IPC, da dies die geringste Distanz erzielt. Im Extremfall werden somit alle Tasks eines vollständig seriellen TG auf demselben IPC platziert. Dies ist hingegen nicht der Fall, wenn ein Task auf mehrere Vorgänger-Tasks wartet, welche sich aufgrund ihrer parallelen Abarbeitung nicht auf demselben IPC befinden.

Je nach Beschaffenheit der zu platzierenden TGs, eignen sich die vorgestellten Strategien unterschiedlich gut für ein temperaturreduzierendes oder ein leistungsorientiertes Mapping. Falls das Temperaturprofil eines Systems möglichst ausgeglichen sein soll, eignet sich für einen TG mit wenigen, dafür aufwands- und zeitintensiven Tasks vermutlich ein last- oder temperaturorientierter Ansatz besser als ein distanzorientierter. Falls allerdings die Leistungsfähigkeit im Vordergrund steht, ist letzterer die bessere Wahl. Aus diesem Grund werden im Rahmen der nachfolgenden Untersuchungen drei verschiedene TG-Typen in die durchgeführten Simulationen einbezogen.

4.3.2 Vergleich mit gängigen Konzepten und Bewertung

Die in diesem Abschnitt durchgeführten Untersuchungen bezüglich der Auswirkung eines proaktiven temperaturorientierten Task-Mappings auf die Leistungsfähigkeit und das Temperaturprofil NoC-basierter Systeme werden für verschiedene TG-Typen durchgeführt. Damit soll gezeigt werden, dass verschiedene Mapping-Strategien sich unterschiedlich gut für bestimmte Anwendungstypen eignen. Diesbezüglich spielt auch die

Zielstellung des Mappings eine bedeutende Rolle, da verschiedene Optimierungsstrategien sich unterschiedlich auf das Systemverhalten auswirken.

Die Parameter der Simulation sind identisch zu den in Tabelle 3.3 aufgeführten Einstellungen. Sämtliche Untersuchungen werden auch hier für ein 4×4 NoC durchgeführt, wobei die Dauer der simulierten Laufzeit 1 s beträgt. Die Simulationsperiode $T_{S,Mapping}$ wird mit 10 ns vergleichsweise niedrig angesetzt, um auf diese Weise die Aktualisierungsrate des Vorhersageprofils auf ein Maximum zu erhöhen. Dies ist von großer Bedeutung für die Effizienz des proaktiven Task-Mappings, da dessen Entscheidungen für das Mapping direkt an dieses Profil gekoppelt sind. Eine hohe Aktualisierungsrate ermöglicht dabei eine wesentlich schnellere Berücksichtigung eines Task-Mappings im Profil und erzeugt somit realistischere Prognosen. Dies gilt auch für das Entfernen bereits fertiggestellter Tasks aus dem Profil, da ein zu langes Verbleiben abgelaufener Tasks das Vorhersageprofil verzerren würde. Generell erhöht ein kleiner Wert für $T_{S,Mapping}$ sowohl die Präzision des Vorhersageprofils als auch dessen Aktualität.

Betrachtete Anwendungstypen Die Anwendungstypen werden durch drei TGs repräsentiert, welche mithilfe von TGFF generiert wurden und sich in markanten Punkten unterscheiden. Im Detail betrifft dies die Anzahl der zu einem TG gehörigen Tasks sowie die Ausführungszeit und die Menge der während dieser Dauer verarbeiteten Daten und durchgeführten Berechnungen. Letzteres wird im Folgenden als Berechnungsaufwand bezeichnet. Darüber hinaus spielt das Verkehrsaufkommen eine große Rolle, welches durch den Datenaustausch zwischen einzelnen Tasks verursacht wird. Die drei genutzten TG-Typen sind nachfolgend aufgeführt:

- **K1:** große Anzahl von Tasks; kurze Ausführungszeit, geringer Berechnungsaufwand und geringes Verkehrsaufkommen je Task
- **K2:** kleine Anzahl von Tasks; lange Ausführungszeit, hoher Berechnungsaufwand und hohes Verkehrsaufkommen je Task
- **K3:** Mischung aus K1 und K2

K1 repräsentiert aufgrund seiner Eigenschaften damit Anwendungen, welche sich in eine Vielzahl kleinerer Teilaufgaben unterteilen lassen. Diese Teilaufgaben sind weniger berechnungs- und kommunikationsintensiv. Solche Anwendungen profitieren damit stark vom Prinzip der Parallelisierung. Bezuglich dieses Anwendungstyps kann das proaktive Task-Mapping seine Vorteile voraussichtlich nicht vollständig ausspielen. Vor allem die kurzen Ausführungszeiten verhindern vermutlich eine zufriedenstellende Abschätzung der Entwicklung des Temperaturprofils. Auch das geringe zu erwartende Verkehrsaufkommen erschwert eine Vorhersage.

K2 hingegen bildet berechnungsintensive Anwendungen ab, welche aus nur wenigen, dafür aber aufwändigen und zeitintensiven Tasks bestehen. Eine Parallelisierung der Teilaufgaben ist für solche Anwendungen nur bedingt möglich, da zudem große Datenmengen zwischen den Tasks ausgetauscht werden müssen. Für diesen Anwendungstyp ist das proaktive Task-Mapping wesentlich besser geeignet. Die hohen Werte für die Ausführungszeit sowie ein hoher Kommunikations- und Berechnungsaufwand begünstigen zuverlässige und kurzfristige Prognosen sowie längerfristige Vorhersagen.

Der Anwendungstyp K3 stellt eine Mischung aus K1 und K2 dar und sollte sich im Vergleich mit den anderen Mapping-Strategien dementsprechend auf die Effizienz des proaktiven Ansatzes auswirken. Mithilfe dieser verschiedenen TG-Typen ist das Spektrum möglicher Anwendungen selbstverständlich längst nicht erschöpft. Allerdings werden zwei grundlegend verschiedene Typen sowie ein ausgewogenes Mittelmaß betrachtet. Damit sollten die Stärken und Schwächen der unterschiedlichen Algorithmen in ausreichendem Maß berücksichtigt werden.

Temperatur

Analog zum Temperaturmanagement ist auch die Auswirkung der Mapping-Strategie auf das Temperaturprofil und das Temperaturgleichgewicht des betrachteten NoC-basierten Systems der entscheidende Faktor. Um zu überprüfen, inwieweit die proaktive temperaturorientierte Strategie Vorteile gegenüber den Vergleichsstrategien bietet, werden die durchschnittliche sowie die Maximaltemperatur des Systems ausgewertet. Zudem wird der maximale Temperaturunterschied zwischen den Systemkomponenten herangezogen, um Aussagen bezüglich der Temperaturbalance zu treffen. Sämtliche Werte stellen dabei Durchschnittswerte dar, welche über die gesamte Simulationsdauer gebildet werden. In Abbildung 4.15 sind die Ergebnisse für die unterschiedlichen Mapping-Strategien unter Verwendung der vorgestellten TG-Typen K1, K2 und K3 illustriert.

Durchschnittliche Temperatur Im Vergleich zu den übrigen Strategien erzielt das proaktive Task-Mapping entgegen der Erwartungen für K1 eine Reduzierung der durchschnittlichen Temperatur. Dies gilt ebenso für die reaktive Strategie. Dieser Sachverhalt deutet darauf hin, dass ein temperaturbasiertes Mapping für aus vielen kleineren Teilaufgaben bestehende Anwendungen nicht geeignet ist. Diese Bewertung ist der Tatsache geschuldet, dass die Senkung der durchschnittlichen Temperatur nur mit Einbußen bezüglich der Leistungsfähigkeit erkauft werden kann (siehe Abbildung 4.16). Da im Grunde lediglich verschiedene Positionierungen einzelner Tasks vorgenommen werden, kann dies theoretisch also nicht zu einer Reduzierung der Gesamtaktivität und damit der insgesamt abgestrahlten Wärme führen ohne die Leistungsfähigkeit zu beeinträchtigen. Auf diesen Punkt wird im weiteren Verlauf genauer eingegangen.

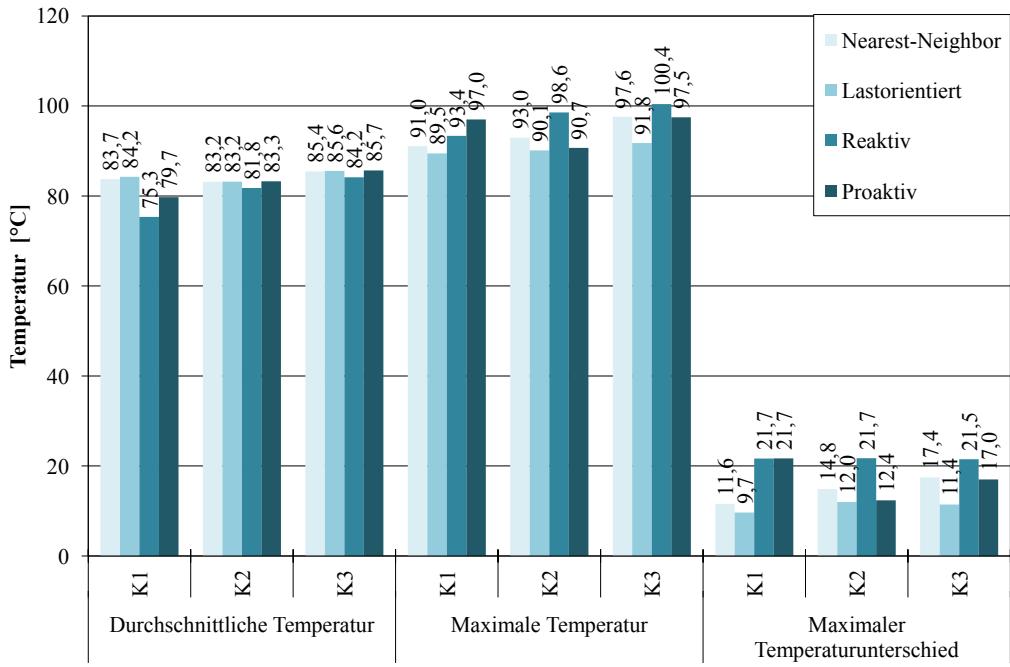


Abbildung 4.15: Ergebnisse bezüglich der Durchschnittstemperatur, der Maximaltemperatur und der maximalen Temperaturunterschiede in einem 4×4 NoC unter Verwendung der untersuchten Mapping-Strategien für verschiedene Task-Graph-Typen

Zu erklären ist dieses Verhalten mit der Beschaffenheit der TGs für K1. Deren Tasks weisen eine kurze Ausführungszeit und einen geringen Berechnungsaufwand auf, was eine adäquate Vorhersage des thermischen Verhaltens zu verhindern scheint. Dies resultiert in Mapping-Entscheidungen, welche offenbar in der Lage sind die Temperatur positiv zu beeinflussen. Die Tasks werden dabei aber derart platziert, dass eine zügige Abarbeitung behindert wird. Der lastorientierte sowie der Nearest-Neighbor-Algorithmus liefern stets annähernd identische Ergebnisse. Dies ist damit zu begründen, dass es aufgrund der großen Anzahl zu platzierender Tasks nicht primär wichtig ist, ob die Last möglichst gleichmäßig verteilt oder die Kommunikationswege minimal gehalten werden. Für K3 ist aufgrund der gemischten Zusammensetzung der TGs kein Algorithmus in der Lage sich von den anderen deutlich abzusetzen.

Maximaltemperatur und maximaler Temperaturunterschied Bezuglich der maximalen Temperatur verhält sich die proaktive Strategie hingegen wie erwartet und erzielt für K2 eine deutliche Verbesserung. Lediglich der lastorientierte Ansatz erreicht ein minimal besseres Ergebnis. Dies zeigt, dass für TGs mit wenigen aufwändigen Tasks (d. h. hoher Berechnungsaufwand, lange Ausführungszeit und große Datenmengen) sowohl ein lastorientiertes als auch ein proaktives temperaturorientiertes Task-Mapping in der Lage sind, das Temperaturprofil des zugrunde liegenden Systems auszubalancieren. Für

eine weitere Verbesserung mithilfe des proaktiven Mappings ist eine Erweiterung der Fähigkeiten bezüglich einer Langzeitprognose erforderlich.

Für K1 schneiden die Vergleichsstrategien besser ab als die proaktive Strategie. Die hohe Maximaltemperatur deutet auf eine Task-Ballung hin und ist auf eine nicht ausreichend präzise Prognose zurückzuführen, welche durch die Task-Eigenschaften verursacht wird. Die besten Resultate erzielt auch hier der lastorientierte Ansatz, da eine Task-Ballung für diesen ausgeschlossen ist.

Dies trifft auch auf K3 zu, wobei der proaktive Algorithmus für diesen TG-Typus im Vergleich zu den Resultaten für K1 deutlich zuverlässigere Prognosen treffen kann. Zurückzuführen ist dies unter anderem auf den gemischten Task-Pool. Dieser enthält neben den sich negativ auf die Vorhersage auswirkenden Tasks des Typs K1 auch besser prognostizierbare Tasks des Typs K2.

Die Tendenz dieser Ergebnisse lässt sich auf den maximalen Temperaturunterschied zwischen den Systemkomponenten übertragen. Das proaktive Task-Mapping spielt auch diesbezüglich seine Vorteile für Tasks mit langen Ausführungszeiten und hohem Berechnungs- und Kommunikationsaufwand aus. Für diese ist das proaktive Mapping in der Lage die Tasks so anzuordnen, dass ein ausgeglicheneres Temperaturprofil entsteht.

Leistungsfähigkeit des Systems

Da alle Mapping-Strategien für die Platzierung der Tasks eines TG verschiedene Kriterien nutzen, treten dementsprechende Abweichungen bezüglich der Leistungsfähigkeit des Systems auf. Anders als für die Bewertung des temperaturorientierten Managements können an dieser Stelle Kriterien wie die Datenrate oder die Dauer einer Paketübertragung nicht herangezogen werden, da diese nur wenig Aufschluss über die Geschwindigkeit der Fertigstellung abzuarbeitender TGs geben. Stattdessen werden für die Bewertung der Effizienz der Strategien die Anzahl der während der Simulation fertiggestellten Tasks sowie die durchschnittliche Leerlaufzeit der IPCs genutzt.

Die Leerlaufzeit eines IPC gibt an, wie lange dieser während der Simulation in seiner internen Warteschlange keinen Task zum Abarbeiten zur Verfügung hatte. Sie ist ein Maß dafür, inwiefern ein Mapping-Algorithmus in der Lage ist, alle IPCs gleichmäßig mit Aufgaben zu versorgen, sodass keine vermeidbaren Phasen der Untätigkeit entstehen. In Abbildung 4.16 sind die auf die proaktive Strategie normierten Ergebnisse für die betrachteten Parameter unter Verwendung der drei TG-Typen dargestellt.

Prinzipiell ähneln die Resultate denen der erzielten durchschnittlichen Temperatur (siehe Abbildung 4.15). Das heißt, dass mit einer Reduzierung der durchschnittlichen Temperatur in jedem Fall mit einer Verringerung der Anzahl fertiggestellter Tasks zu rechnen ist. Wie bereits angedeutet wurde, ist dies damit zu begründen, dass durch eine reine Umverteilung einzelner Tasks lediglich der Ort der Wärmeausbreitung, jedoch

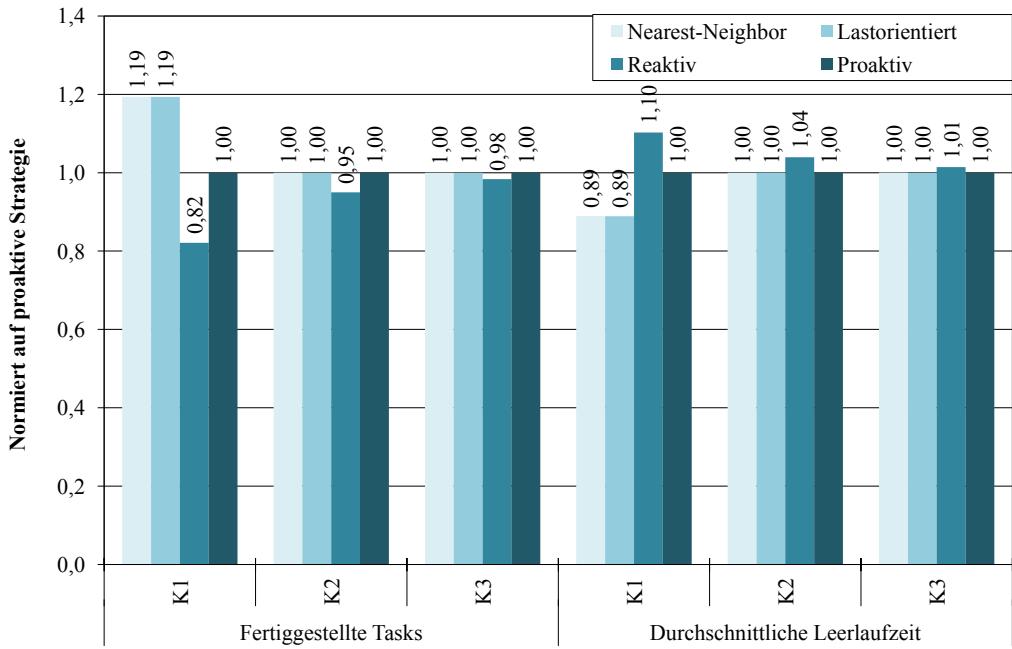


Abbildung 4.16: Anzahl fertiggestellter Tasks und durchschnittliche Leerlaufdauer der Systemkomponenten in einem 4×4 NoC unter Verwendung der untersuchten Mapping-Strategien für verschiedene TG-Typen (normiert auf die proaktive Strategie)

nicht die Wärmemenge variiert werden kann. Dies beeinflusst lediglich die Maximaltemperatur sowie den maximalen Temperaturunterschied. Wird dennoch die durchschnittliche Temperatur reduziert, so kann dies nur durch eine verringerte Anzahl von Tasks und damit durch eine geringere Wärmemenge hervorgerufen werden. Mit einer Verringerung der Anzahl fertiggestellter Tasks erhöht sich automatisch auch die Leerlaufzeit der IPCs. Das proaktive Task-Mapping reduziert lediglich für K1 die Anzahl fertiggestellter Tasks und verringert damit ebenfalls die Durchschnittstemperatur. Dies spiegelt sich auch in der Leerlaufzeit für K1 wider. Für die TG-Typen K2 und K3 kann die volle Leistungsfähigkeit aufrecht erhalten werden.

Die Ergebnisse für K1 sind damit zu erklären, dass die Tasks durch die proaktive und die reaktive Strategie aus einer leistungsorientierten Perspektive vergleichsweise ungünstig platziert werden. Dies bremst die Kommunikation untereinander abhängiger Tasks aus und verzögert deren Fertigstellung, sodass insgesamt weniger Tasks abgearbeitet werden. Eine weitere Folge ist, dass IPCs, deren Task bereits beendet ist, unter Umständen auf die Fertigstellung eines anderen Tasks desselben TG warten müssen, um fortfahren zu können. Vor diesem Hintergrund ist eine Temperaturreduzierung auf Kosten der Leistungsfähigkeit nicht wünschenswert, da anders als beim Temperaturmanagement keine Reduzierung der Leistungsfähigkeit einkalkuliert ist. Die vorhandenen Tasks sollen stattdessen ohne negative Auswirkungen möglichst effizient platziert werden.

Damit erscheinen die temperaturbezogenen Parameter der Maximaltemperatur und des maximalen Temperaturunterschieds von deutlich höherem Interesse zu sein. Konzentriert man die Betrachtungen im Zusammenhang mit der Leistungsfähigkeit auf diese, so ergibt sich erneut die Schlussfolgerung, dass sich das proaktive Task-Mapping für wenige Tasks mit langen Ausführungszeiten und hohem Bearbeitungsaufwand sowie hohem Verkehrsaufkommen am besten eignet. Im Hinblick auf die Leistungsfähigkeit gilt dies neben K2 auch für K3, da auch für diesen TG-Typus keinerlei Einbußen zu verzeichnen sind. Ein Abgleich mit den temperaturbezogenen Parametern verweist allerdings darauf, dass im Vergleich zu anderen Mapping-Strategien nur für TGs vom Typ K2 deutliche Reduzierungen der Maximaltemperatur sowie des maximalen Temperaturunterschieds erreicht werden.

4.3.3 Zusammenfassung

In diesem Abschnitt wurde mit einem temperaturorientierten proaktiven Task-Mapping eine weitere Anwendungsmöglichkeit für das Temperaturmodell vorgestellt, welches im Rahmen dieser Arbeit entwickelt wurde. Im Gegensatz zum proaktiven Temperaturmanagement, welches ausgehend von Aktivitätsüberwachungen mithilfe einer Temperaturvorhersage dynamische Leistungsanpassungen vornimmt, wird das Modell an dieser Stelle für eine temperaturgebundene Platzierung einzelner Tasks zur Laufzeit NoC-basierter Systeme eingesetzt. Dabei wird anhand der Eigenschaften eines Tasks die Auswirkung der Platzierung desselben auf die Temperaturentwicklung prognostiziert. Mit dieser Information lassen sich dann nachfolgende Tasks derart platzieren, dass ein möglichst ausgewogenes Temperaturprofil entsteht.

Während der durchgeführten Untersuchungen wurde deutlich, dass sich das Konzept in seiner derzeitigen Form besonders für bestimmte Anwendungen eignet. Dies sind vor allem Anwendungen, welche typischerweise aus verhältnismäßig aufwändigen Tasks bestehen. Solche Tasks weisen günstigstenfalls eine lange Ausführungsduer sowie einen hohen Berechnungsaufwand und ein hohes Verkehrsaufkommen auf, welches durch die Kommunikation mit anderen Tasks entsteht. Sind diese Voraussetzungen erfüllt, ist das proaktive Task-Mapping im Vergleich zu herkömmlichen Mapping-Strategien in der Lage die Maximaltemperatur sowie die maximal auftretenden Temperaturunterschiede im System zu reduzieren. Gleichzeitig wird die Leistungsfähigkeit des Systems für solche Task-Typen in keiner Weise beeinträchtigt. Damit lassen sich ohne Einbußen Temperaturspitzen und Temperaturungleichgewichte reduzieren, indem einzelne Tasks vorausschauend platziert werden.

Des Weiteren zeigt sich, dass sich die erzielten Verbesserungen nicht auf beliebige Anwendungstypen übertragen lassen. Besteht ein TG beispielsweise aus einer Vielzahl schnell abzuarbeitender Tasks, so ist das proaktive Task-Mapping nicht in der Lage, sei-

Tabelle 4.6: Qualitative Bewertung des Einflusses des proaktiven Task-Mappings auf Temperatur und Leistungsfähigkeit für verschiedene Anwendungstypen (Legende: + positiv, \circ mittel, - negativ, \times neutral)

Typ	T_{\emptyset}	T_{Max}	ΔT_{Max}	Leistung
K1	\times	-	-	\circ
K2	\times	+	+	+
K3 = K1 + K2	\times	\circ	\circ	\times

nen positiven Einfluss geltend zu machen. Dies ist vor allem für Tasks der Fall, welche eine kurze Ausführungsduer sowie einen geringen Berechnungsaufwand aufweisen und ein geringes Verkehrsaufkommen verursachen. Besonders die kurzen Ausführungszeiten der Tasks scheinen in diesem Zusammenhang eine verlässliche Vorhersage der Temperaturentwicklung zu erschweren.

Das Fazit der durchgeführten Betrachtungen lautet daher, dass ein proaktives temperaturorientiertes Task-Mapping in der Lage ist das thermische Verhalten NoC-basierter Mehrkernprozessoren positiv zu beeinflussen und dabei deren Leistungsfähigkeit nicht zu beeinträchtigen. Diese Fähigkeit ist jedoch an die auf dem System ausgeführten Anwendungen gebunden.

Eine qualitative Bewertung des proaktiven Task-Mappings in Kombination mit den Anwendungstypen K1, K2 und K3 (siehe Abschnitt 4.3.2) wird in Tabelle 4.6 gegeben. Die durchschnittliche Temperatur T_{\emptyset} kann theoretisch für keinen der Typen beeinflusst werden. Abweichende Platzierungen einzelner Tasks führen zwar zu verschiedenen Temperaturprofilen, jedoch kann die insgesamt abgegebene Wärme nicht verringert werden. T_{\emptyset} muss demnach konstant bleiben. Andernfalls würde unzulässigerweise eine geringere Anzahl von Tasks platziert. T_{Max} und ΔT_{Max} können hingegen besonders für zeitintensive und aufwändige Anwendungstypen (d. h. K2) positiv beeinflusst werden. Dies gilt bis zu einem gewissen Grad auch für gemischte Anwendungen (d. h. K3).

Die Leistungsfähigkeit kann jedoch nur für Anwendungen vom Typ K2 aufrecht erhalten werden. Gemischte Anwendungen erzeugen durch ihre Beschaffenheit eine gewisse Unschärfe, sodass sowohl der Einfluss unterschiedlicher Mapping-Algorithmen als auch die Auswirkung verschiedener Anwendungstypen verschwimmt.

5 Zusammenfassung und Ausblick

5.1 Zusammenfassung

Den stetig wachsenden Anforderungen an hochintegrierte Schaltungen bezüglich Leistungsfähigkeit, Dimensionierung und Flexibilität konnte bislang mit einer Skalierung der Halbleitertechnologien begegnet werden. Die mithilfe dieser Strategie erzielten Erfolge basieren darauf, dass mit jeder Technologiegeneration eine größere Transistorzahl auf gleicher Fläche integriert werden konnte. Aktuelle Fertigungstechnologien mit Strukturgrößen im Bereich weniger Nanometer stoßen jedoch zunehmend an physikalische Grenzen. Neben leistungsbezogenen Parametern müssen für den Schaltungsentwurf daher weitere Faktoren wie Leistungsaufnahme und Zuverlässigkeit berücksichtigt werden.

Insbesondere die Zuverlässigkeit gewinnt stark an Bedeutung, da beispielsweise zunehmende Leckströme und steigende Leistungsdichten zu einer erhöhten Defektanfälligkeit digitaler Schaltungen führen und das Auftreten von Alterungs- und Verschleißerscheinungen beschleunigen. Diesbezüglich rücken auch thermische Aspekte immer mehr in den Vordergrund, da höhere Integrationsdichten mit einer steigenden Dichte der thermischen Verlustleistung einhergehen. Dies führt neben thermischem Stress zusätzlich dazu, dass diverse physikalische Effekte begünstigt werden, welche die Alterung digitaler Schaltungen beschleunigen.

Darüber hinaus führt die trotz neuer Fertigungstechnologien nur begrenzt steigerbare Leistungsfähigkeit einzelner Systemkomponenten dazu, dass der geforderte Leistungszuwachs durch ein wachsendes Maß an Parallelität erreicht werden muss. Dabei werden mehrere Aufgaben zeitgleich durch verschiedene, miteinander kommunizierende Systemkomponenten bearbeitet. Die damit einhergehende Verlagerung einer seriell arbeitenden, berechnungszentrierten Architektur hin zu einem parallel arbeitenden, kommunikationsorientierten Entwurf gipfelt in der Entstehung sogenannter Multi-Processor-Systems-on-Chip (MPSoCs).

Aufgrund der erhöhten Systemkomplexität und eines steigenden Kommunikationsaufkommens werden die einzelnen Komponenten mithilfe von Networks-on-Chip (NoC) verbunden. Diese zeichnen sich durch eine gute Skalierbarkeit sowie eine hohe Leistungsfähigkeit aus und sind in der Lage sowohl die technologischen als auch die architekturellen Problemstellungen zu überwinden und damit die Leistungsanforderungen moderner MPSoCs zu erfüllen. Der gesteigerte Verwaltungs- und Koordinationsaufwand

solcher Systeme ist jedoch nach wie vor problematisch und bedarf einer gesonderten Be- trachtung.

Der dargelegte Sachverhalt betont besonders zwei Aspekte, die im Rahmen dieser Arbeit bearbeitet wurden. Einerseits betrifft dies den durch Temperaturspitzen und Temperaturungleichgewichte entstehenden thermischen Stress. Durch dessen Reduzierung sollen temperaturbedingte Alterungs- und Verschleißerscheinungen hinausgezögert werden. Das frühzeitige Erkennen von Temperaturtrends zur Vorbeugung unerwünschter Temperaturverteilungen erscheint dabei als vielversprechender Ansatz. Andererseits liegt der Fokus dabei auf NoC-basierten Architekturen, sodass bei der Entwicklung entsprechender Konzepte unter anderem deren verteilter Charakter sowie eine erhöhte Systemkomplexität berücksichtigt werden müssen. Die vorliegende Arbeit liefert diesbezüglich wertvolle Erkenntnisse über die Realisierbarkeit proaktiver Maßnahmen zur Reduzierung des thermischen Stresses und stellt entsprechende Konzepte für NoC-basierte Mehrkernprozessoren vor.

Zu diesem Zweck wurde zunächst die technologische Entwicklung digitaler Schaltungen betrachtet. Dabei konnten neben bestehenden Problemstellungen zusätzlich die Integrität sowie die Lebensdauer hochintegrierter Schaltungen als zunehmend kritische Entwurfs- und Betriebsparameter identifiziert werden. Des Weiteren wurde deutlich, dass stets Kompromisse nötig sind, um leistungsfähige und dennoch zuverlässige und energieeffiziente Systeme zu realisieren.

In diesem Zusammenhang wurden insbesondere die Ursachen für das Fehlverhalten hochintegrierter Schaltungen näher betrachtet. Neben thermischem Stress wurde eine Reihe temperaturabhängiger physikalischer Effekte herausgearbeitet, welche besonders zu Verschleiß- und Ausfallerscheinungen beitragen und damit langfristig die Zuverlässigkeit digitaler Systeme gefährden. Mithilfe des Arrhenius-Modells konnte nachgewiesen werden, dass die Geschwindigkeit solcher Effekte mit steigender Temperatur zunimmt. Hohe Temperaturen verkürzen somit die Lebensdauer digitaler Schaltungen nachweisbar.

Des Weiteren wurden anhand eines Überblicks über die Kommunikationsstrukturen nanoelektronischer Systeme die Vorteile NoC-basierter Systeme dargelegt. Beispielhafte Untersuchungen bezüglich des Ausfalls einzelner, vergleichsweise simpler Kommunikationskomponenten machten deutlich, dass selbst hochgradig vernetzte NoC-Architekturen Ausfälle nicht in akzeptablem Umfang kompensieren können. Im Allgemeinen erscheint eine Verhinderung beziehungsweise Verzögerung von Ausfällen ohnehin als die praktikablere Lösung, da die Umgehung defekter Bereiche (z. B. mithilfe adaptiver Routing-Algorithmen) eine thermische Mehrbelastung für die verbliebenen Komponenten bedeuten würde. Da viele der verschleiß- und alterungsrelevanten Faktoren temperaturabhängig sind, erscheint diesbezüglich eine dynamische Regelung der Leistungsfä-

higkeit zur proaktiven Beeinflussung der Temperaturentwicklung als lohnenswertester Ansatz. Dass die für eine proaktive Steuerung benötigte Verzögerung zwischen Schaltvorgängen und einer entsprechenden Temperaturerhöhung ausreichend groß ist, konnte unter Zuhilfenahme eines FPGA-basierten Versuchsaufbaus nachgewiesen werden.

Weiterhin wurde ein Temperaturmodell vorgestellt, welches für die proaktiven Konzepte eines Systemmanagements benötigt wird. Dieses greift auf den Ansatz zurück, das thermische Verhalten hochintegrierter Schaltkreise mithilfe äquivalenter elektrischer RC-Glieder zu modellieren. Das Modell wurde in eine bereits vorhandene SystemC-basierte Umgebung zur funktionalen Simulation NoC-basierter Mehrkernprozessoren integriert. Damit ist es möglich die Temperaturentwicklung NoC-basierter Systeme auf Basis registrierter Schaltaktivitäten zur Laufzeit zu modellieren und dieses Wissen für eine proaktive Beeinflussung der Temperatur zu nutzen. Dass dies mit ausreichender Geschwindigkeit bei gleichzeitig angemessener Genauigkeit möglich ist, konnte in entsprechenden Simulationen gezeigt werden. Ein Kompromiss zwischen Genauigkeit und Geschwindigkeit ist dennoch unvermeidlich und muss sich den Anforderungen an das jeweilige Managementsystem beugen.

Nicht zuletzt die zwei gezeigten Anwendungsszenarien für das entwickelte Temperaturmodell konnten dessen praktische Verwendbarkeit untermauern. Zum Einen betrifft dies den Einsatz für ein proaktives Temperaturmanagement zur Laufzeit NoC-basierter Systeme. Die Simulationsumgebung wurde um ein entsprechendes verteiltes Überwachungssystem sowie eine zentrale Steuereinheit erweitert. Das Überwachungssystem besteht im Einzelnen aus Aktivitätszählern, welche jeder Systemkomponente zugewiesen werden und der Steuereinheit detektierte Aktivitäten melden. Diese wiederum prognostiziert auf Basis der gemeldeten Aktivitätsstatistiken die voraussichtliche Temperaturentwicklung und leitet entsprechende Maßnahmen ein.

Den größten Vorteil stellen die im Vergleich zu reaktiven Konzepten verkürzten Antwortzeiten auf unerwünschte Temperaturentwicklungen dar. Dieser Zeitvorteil lässt sich entweder nutzen, um bei gleichem Aufwand eine Reduzierung der Temperatur zu erzielen oder um den Aufwand für identische Resultate zugunsten einer erhöhten Leistungsfähigkeit zu reduzieren. Für die Bewertung der Effizienz des proaktiven Konzepts wurde ein reaktiver Ansatz herangezogen, bei welchem das Überwachungssystem anstelle von Schaltaktivitäten Temperaturwerte überwacht und diese an die Steuereinheit weiterleitet.

Aus den durchgeführten Untersuchungen bezüglich der Leistungsfähigkeit und des Temperaturverhaltens konnten zwei wichtige Erkenntnisse gewonnen werden. Einerseits ist unabhängig von der eingesetzten Managementstrategie immer ein Kompromiss zwischen der Reduzierung des thermischen Stresses und den damit verbundenen Leistungseinbußen erforderlich. Andererseits zeigte sich im Vergleich, dass das proaktive Tem-

raturmanagement diesbezüglich über sehr viel mehr Spielraum verfügt. Damit konnten durch entsprechende Konfigurationen entweder die Leistungsdefizite bei teilweiser Aufrechterhaltung des positiven Einflusses auf die Temperaturverteilung verringert werden. Oder der thermische Stress konnte wesentlich deutlicher reduziert werden.

Das zweite Anwendungsszenario sah eine Verwendung des Temperaturmodells für ein proaktives Task-Mapping zur Laufzeit NoC-basierter Systeme vor. Im Zuge dessen wurde die Simulationsumgebung um die Modellierung von Task-Graphen erweitert. Auf diese Weise konnte die Abarbeitung von Anwendungen durch das System simuliert werden, welche aus mehreren Teilaufgaben bestehen. Bevor der Task eines Task-Graphen auf einer Systemkomponente platziert wird, wird durch den proaktiven Mapping-Algorithmus die Auswirkung dieser Platzierung auf die Temperaturentwicklung prognostiziert. Um diese Auswirkung möglichst genau abschätzen zu können, werden die Ausführungszeit, der Bearbeitungsaufwand, der zu erwartende Kommunikationsaufwand sowie alle weiteren beteiligten IPCs in die Prognose einbezogen. Mithilfe der Prognose ist es dem Algorithmus möglich, die nachfolgenden Tasks auf den voraussichtlich kühnst Komponenten zu platzieren und damit das Temperaturprofil des Systems auszubalancieren.

Der Vergleich mit einem reaktiven temperaturorientierten sowie einem last- und einem distanzorientierten Mapping-Algorithmus führte zu drei wichtigen Erkenntnissen. Erstens lässt sich die durchschnittliche Temperatur eines Systems durch eine variierende Verteilung der Tasks kaum positiv beeinflussen. Dies kann nur auf Kosten der Leistungsfähigkeit erreicht werden. Zweitens eignet sich die proaktive Strategie, um sowohl die Maximaltemperatur als auch die maximal auftretenden Temperaturunterschiede zu minimieren, ohne die Leistungsfähigkeit zu beeinträchtigen. Dies führt zu einem ausgeglicheneren Temperaturprofil und mindert den thermischen Stress. Drittens ist diese Verbesserung stark von den auf dem System ausgeführten Anwendungen abhängig.

5.2 Ausblick

Das im Rahmen dieser Arbeit entwickelte Temperaturmodell sowie die daraus resultierenden Konzepte für ein temperaturorientiertes Management NoC-basierter Systeme bilden die Grundlage für weitere Forschungen. Darüber hinaus verweisen die Resultate der durchgeführten Untersuchungen auf das bestehende Potential für Verbesserungen der einzelnen Konzepte.

Bezüglich des Temperaturmodells wären Modifikationen wünschenswert, welche die Genauigkeit gering aufgelöster Modelle erhöhen beziehungsweise den Zeitaufwand von Modellen höherer Auflösung deutlich reduzieren. Damit wäre eine exaktere und dennoch leistungsfähige Temperaturmodellierung zur Laufzeit NoC-basierter Systeme

möglich. Zudem gilt es zu untersuchen, inwieweit dies den proaktiven Managementkonzepten zuträglich wäre.

Des Weiteren ergaben die Untersuchungen, dass ein Großteil der insgesamt benötigten Zeit für die Erstellung und Vorbereitung des Temperaturmodells aufgewendet wird. Dieser Teil der Modellierung ist von der eigentlichen Simulation entkoppelt. Außerdem hängt das in Form einer Differentialgleichung vorliegende Modell lediglich von geometrischen und architekturbezogenen Parametern ab. Daher wäre eine Auslagerung der Vorbereitungsphase erstrebenswert, da somit die Zeit bis zum Beginn der eigentlichen Simulation insbesondere für aufwändige Modelle drastisch verkürzt werden könnte. Diesbezüglich würde sich beispielsweise die Erstellung einer Bibliothek vorkompilierter Temperaturmodelle verschiedener NoC-basierter Systeme anbieten, sodass lediglich das korrekte System ausgewählt werden müsste. Das größte Hindernis für die Umsetzung dieses Konzepts stellt die Speicherung der vorbereiteten Modelle dar.

Da die durch das proaktive Temperaturmanagement erzielbaren Verbesserungen des thermischen Verhaltens unumgänglich mit einer Reduzierung der Leistungsfähigkeit NoC-basierter Systeme verbunden sind, sind auch an dieser Stelle Verbesserungen wünschenswert. In diesem Zusammenhang scheint die Realisierung dynamisch anpassbarer Managementparameter am aussichtsreichsten. Auf diese Weise könnte flexibel auf die Bedürfnisse der durch das System aktuell bearbeiteten Anwendungen eingegangen werden. Hoch priorisierte, mit Echtzeitanforderungen behaftete Anwendungen könnten somit ungeachtet der Auswirkung auf die Temperaturentwicklung mit hoher Leistungsfähigkeit bearbeitet werden. Aufgaben niedriger Priorität könnten hingegen zugunsten einer Temperaturreduzierung gezielt verlangsamt abgearbeitet werden, um das Temperaturprofil positiv zu beeinflussen. Falls bereits während der Entwurfsphase die Aufgabengebiete der einzelnen Systemkomponenten bekannt sind, könnte dieser Ansatz auch statisch mit speziell angepassten Direktiven bezüglich der Managementparameter realisiert werden. Ähnliches gilt auch für das Konzept des proaktiven Task-Mappings, nur das für dieses eine dynamische, situationsgebundene Wahl des vorteilhaftesten Mapping-Algorithmus denkbar wäre. Eine Wahl in Abhängigkeit des Typus der zu platzierenden Anwendung scheint dabei am erfolgversprechendsten.

Ein weiterer Aspekt, der im Rahmen dieser Arbeit nicht erschöpfend behandelt werden konnte, ist die Frage nach der langfristigen Wirkung reduzierter Temperaturen und ausgeglichener Temperaturprofile. Im Einzelnen wäre unter anderem zu klären in welchem Umfang die mithilfe der entwickelten Konzepte erzielten Verbesserungen tatsächlich zu einer Verlängerung der Lebensdauer hochintegrierter Schaltungen und damit zu einer Erhöhung der Zuverlässigkeit NoC-basierter Systeme beitragen. Hierfür sind Langzeitsimulationen repräsentativer Systeme erforderlich.

Tabelle 5.1: Abschätzung der Auswirkung einer Kombination der proaktiven Konzepte für Temperaturmanagement und Task-Mapping auf die Temperatur und die Leistungsfähigkeit (Legende: PT - Proaktives Temperaturmanagement, PM - Proaktives Mapping, RT- Reaktives Temperaturmanagement)

	T_{\emptyset}	T_{Max}	ΔT_{Max}	Leistung
Kein	$T_{\emptyset,K}$	$T_{Max,K}$	$\Delta T_{Max,K}$	X
Reaktiv: RT	$T_{\emptyset,K} - \Delta T_{RT} + \epsilon T_{RT}$	$T_{Max,K}$	$\Delta T_{Max,K}$	$X - \Delta X_{RT}$
Proaktiv: PT	$T_{\emptyset,K} - n \cdot \Delta T_{RT} + \epsilon T_{PT}$	$T_{Max,K} - n \cdot \Delta T_{PT}$	$\Delta T_{Max,K} - n \cdot \Delta T_{PT}$	$X - \Delta X_{RT} - f(n, \Delta T_{PT})$
Proaktiv: PT + PM	$T_{\emptyset,K} - n \cdot \Delta T_{RT} + \epsilon T_{PT} + \epsilon T_{PM}$	$T_{Max,K} - n \cdot \Delta T_{PT} - \Delta T_{PM}$	$\Delta T_{Max,K} - n \cdot \Delta T_{PT} - \Delta T_{PM}$	$X - \Delta X_{RT} - f(n, \Delta T_{PT})$

Tabelle 5.1 gibt einen Ausblick bezüglich der Auswirkung einer kombinierten Anwendung der proaktiven Ansätze für Temperaturmanagement und Task-Mapping und bezieht sich dabei auf die qualitativen Bewertungen der jeweiligen Konzepte (siehe Tabellen 4.5 und 4.6).

Es wird erwartet, dass die durch ein proaktives Temperaturmanagement erzielbare Reduzierung des thermischen Stresses insbesondere für die maximale Temperatur T_{Max} sowie den maximalen Temperaturunterschied ΔT_{Max} ausgebaut werden kann. Der dafür zusätzlich nötige Aufwand schlägt sich in Form von ϵT_{PM} negativ auf die durchschnittliche Temperatur T_{\emptyset} nieder, wird aber als vergleichsweise gering angenommen.

Darüber hinaus ist durch den Zusatz des proaktiven Task-Mappings keine nennenswerte Reduzierung der Leistungsfähigkeit des Systems zu erwarten. Dies setzt geeignete, durch das System bearbeitete Anwendungen voraus, welche sich durch lange Ausführungszeiten sowie einen hohen Bearbeitungsaufwand auszeichnen.

Literaturverzeichnis

- [ABG12] A. Aloisio, V. Bocci, and R. Giordano. Rad-tolerance tests of FPGAs and SerDeses for SuperB, 2012. *3rd SuperB Collaboration Meeting*.
- [Ada12a] Adapteva Epiphany-IV 64-Core microprocessor, abgerufen am 11.10.2012. <http://www.adapteva.com/products/silicon-devices/e64g401/>.
- [Ada12b] Adapteva eMesh network-on-chip: Epiphany architecture reference manual, abgerufen am 12.10.2012. <http://www.adapteva.com/support/docs/e3-reference-manual/>.
- [AE03] M. Anis and M. I. Elmasry. *Multi-Threshold CMOS Digital Circuits: Managing Leakage Power*. Springer, 2003.
- [AEK06] B. Ahmad, A. T. Erdogan, and S. Khawam. Architecture of a dynamically reconfigurable NoC for adaptive reconfigurable MPSoC. In *Proceedings of the 1st NASA/ESA conference on Adaptive Hardware and Systems*, pages 405–411, 2006.
- [AM09] D. Atienza and E. Martinez. Inducing thermal-awareness in multicore systems using networks-on-chip. In *VLSI. IEEE Computer Society Annual Symposium on*, pages 187–192, 2009.
- [AMS] SystemC AMS extensions user's guide.
- [AMS13] Accellera SystemC Analog Mixed Signal extensions 1.0, abgerufen am 03.05.2013. <http://www.accellera.org/downloads/standards/systemc/ams/>.
- [Arm13] Arm microprocessor cores, abgerufen am 31.01.2013. <http://www.arm.com/products/processors/index.php>.
- [Art12] Arteris FlexNoC Interconnect IP, abgerufen am 11.10.2012. <http://www.arteris.com/flexnoc>.
- [AV02] D.H. Ahn and J.S. Vetter. Scalable Analysis Techniques for Microprocessor Performance Counter Metrics. In *Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, pages 1–16, 2002.
- [Bau05] R. Baumann. Soft errors in advanced computer systems. *Design Test of Computers, IEEE*, 22(3):258–266, May 2005.
- [BBWW07] E. W. Briao, D. Barcelos, F. Wronski, and F. R. Wagner. Impact of task migration in NoC-based MPSoCs for soft real-time applications. In *Very Large Scale Integration. IFIP International Conference on*, pages 296–299, 2007.

- [BGR⁺97] A. Bravaix, D. Goguenheim, N. Revil, M. Varrot, and P. Mortini. Effects of high temperature on performances and hot-carrier reliability in DC/AC stressed 0.35 μ m n-MOSFETs. In *Solid-State Device Research Conference. Proceeding of the 27th European*, pages 584–587, 1997.
- [Bla04] D.L. Blackburn. Temperature measurements of semiconductor devices - a review. In *Semiconductor Thermal Measurement and Management Symposium. 20th Annual IEEE*, pages 70–80, 2004.
- [BLB97] E. I. Boemo and S. López-Buedo. Thermal monitoring on FPGAs using ring-oscillators. In *Proceedings of the 7th International Workshop on Field-Programmable Logic and Applications*, pages 69–78, 1997.
- [Bor05] S. Borkar. Designing reliable systems from unreliable components: the challenges of transistor variability and degradation. *Micro, IEEE*, 25(6):10–16, November 2005.
- [Bor07] S. Borkar. Thousand core chips: a technology perspective. In *Proceedings of the 44th annual Design Automation Conference*, pages 746–749, 2007.
- [BTM00] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: a framework for architectural-level power analysis and optimizations. In *Computer Architecture. Proceedings of the 27th International Symposium on*, pages 83–94, 2000.
- [C.11] Cornelius C. *Design of complex integrated systems based on networks-on-chip - Trading off performance, power and reliability*. PhD thesis, Institute of Applied Microelectronics and Computer Engineering, University of Rostock, 2011.
- [CBR⁺05] C. Ciordas, T. Basten, A. Rădulescu, K. Goossens, and J. V. Meerbergen. An event-based monitoring service for networks on chip. *ACM Trans. Des. Autom. Electron. Syst.*, 10(4):702–723, October 2005.
- [CC04] H.-C. Chi and J.-H. Chen. Design and implementation of a routing switch for on-chip interconnection networks. In *Advanced System Integrated Circuits. Proceedings of IEEE Asia-Pacific Conference on*, pages 392–395, 2004.
- [CGB97] Y.-S. Chang, S.K. Gupta, and M.A. Breuer. Analysis of ground bounce in deep sub-micron circuits. In *VLSI Test Symposium. 15th IEEE*, pages 110–116, 1997.
- [CGRB06] C. Ciordas, K. Goossens, A. Rădulescu, and T. Basten. NoC monitoring: impact on the design flow. In *Circuits and Systems. Proceedings. IEEE International Symposium on*, pages 1981–1984, 2006.
- [CK95] Z. Chen and I. Koren. Techniques for yield enhancement of VLSI adders. In *Application Specific Array Processors. Proceedings. International Conference on*, pages 222–229, 1995.
- [Cle97] J. J. Clement. Reliability analysis for encapsulated interconnect lines under DC and pulsed DC current using a continuum electromigration transport model. *Journal of Applied Physics*, 82(12):5991–6000, September 1997.

- [CLKK06] J. P. Campbell, P. M. Lenahan, A. T. Krishnan, and S. Krishnan. NBTI: An atomic-scale defect perspective. In *Reliability Physics Symposium Proceedings. 44th Annual., IEEE International*, pages 442–447, 2006.
- [CLP⁺09] F. Clermidy, R. Lemaire, X. Popon, D. Ktenas, and Y. Thonnart. An open and reconfigurable platform for 4G telecommunication: Concepts and application. In *Digital System Design, Architectures, Methods and Tools. 12th Euromicro Conference on*, pages 449–456, 2009.
- [Cor99] B. Cordan. An efficient bus architecture for system-on-chip design. In *Custom Integrated Circuits. Proceedings of the IEEE*, pages 623–626, 1999.
- [CRG08] A. Coskun, T. S. Rosing, and K. C. Gross. Proactive temperature balancing for low cost thermal management in MPSoCs. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 250–257, 2008.
- [CRWG08] A. K. Coskun, T. S. Rosing, K. A. Whisnant, and K. C. Gross. Temperature-aware MPSoC scheduling for reducing hot spots and gradients. In *Proceedings of the Asia and South Pacific Design Automation Conference*, pages 49–54, 2008.
- [CSGS85] K.-L. Chen, S.A. Saller, I.A. Groves, and D.B. Scott. Reliability effects on MOS transistors due to hot-carrier injection. *Solid-State Circuits, IEEE Journal of*, 20(1):306–313, February 1985.
- [CSZ⁺07] P. Chen, M.-C. Shie, Z.-Y. Zheng, Z.-F. Zheng, and C.-Y. Chu. A fully digital time-domain smart temperature sensor realized with 140 FPGA logic elements. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 54(12):2661–2668, December 2007.
- [CYTZ10] M. Chatti, S. Yehia, C. Timsit, and S. Zertal. A hypercube-based NoC routing algorithm for efficient all-to-all communications in embedded image and signal processing applications. In *High Performance Computing and Simulation, International Conference on*, pages 623–630, 2010.
- [dMB06] G. de Micheli and L. Benini. *Networks on Chip: Technology and Tools*. Morgan Kaufmann Publishers Inc., 2006.
- [DN07] B.S. Deepaksubramanyan and A. Nunez. Analysis of subthreshold leakage reduction in CMOS digital circuits. In *Circuits and Systems. 50th Midwest Symposium on*, pages 1400–1404, 2007.
- [DRW98] R. P. Dick, D. L. Rhodes, and W. Wolf. TGFF: Task Graphs For Free, 1998.
- [DT01] W.J. Dally and B. Towles. Route packets, not wires: on-chip interconnection networks. In *Design Automation Conference. Proceedings*, pages 684–689, 2001.
- [DT03] W. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., 2003.

- [DT07] A. Dutta and N.A. Touba. Reliable network-on-chip using a low cost unequal error protection code. In *Defect and Fault-Tolerance in VLSI Systems. 22nd IEEE International Symposium on*, pages 3–11, 2007.
- [ERHH11] T. Ebi, H. Rauchfuss, A. Herkersdorf, and J. Henkel. Agent-based thermal management using real-time I/O communication relocation for 3D many-cores. In *Proceedings of the 21st international conference on Integrated circuit and system design: power and timing modeling, optimization, and simulation*, pages 112–121, 2011.
- [ESA06] ECSS-Q-30-11A, 2006. European Cooperation for Space Standardization.
- [FKCC06] A.P. Frantz, F.L. Kastensmidt, L. Carro, and E. Cota. Dependable network-on-chip router able to simultaneously tolerate soft errors and crosstalk. In *Test Conference. IEEE International*, pages 1–9, 2006.
- [FRD08] J. Flich, S. Rodrigo, and J. Duato. An efficient implementation of distributed routing algorithms for NoCs. In *Networks-on-Chip. Second ACM/IEEE International Symposium on*, pages 87–96, 2008.
- [GDR05] K. Goossens, J. Dielissen, and A. Radulescu. \mathcal{A} ethereal network on chip: concepts, architectures, and implementations. *Design Test of Computers, IEEE*, 22(5):414–421, September 2005.
- [GNR⁺10] L. Guang, E. Nigussie, P. Rantala, J. Isoaho, and H. Tenhunen. Hierarchical agent monitoring design approach towards self-aware parallel systems-on-chip. *ACM Trans. Embed. Comput. Syst.*, 9(3):1–24, March 2010.
- [GPW02] W. H. Gerling, A. Preussger, and F. W. Wulfert. Reliability qualification of semiconductor devices based on physics-of-failure and risk and opportunity assessment. *Quality and Reliability Engineering International*, 18(2):81–98, April 2002.
- [GWG⁺13] M. Gag, T. Wegner, P. Gorski, A. Tockhorn, and D. Timmermann. System level modeling of networks-on-chip for power estimation and design space exploration. In *16. Workshop Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen*, pages 25–34, 2013.
- [GWT10] M. Gag, T. Wegner, and D. Timmermann. System level power estimation of system-on-chip interconnects in consideration of transition activity and crosstalk. In *International Workshop on Power and Timing Modeling, Optimization and Simulation*, pages 21–30, 2010.
- [GWWT12] M. Gag, T. Wegner, A. Waschki, and D. Timmermann. Temperature and on-chip crosstalk measurement using ring oscillators in FPGA. In *15th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems*, pages 201–204, 2012.
- [HAQT⁺04] W. Hung, C. Addo-Quaye, T. Theocarides, Y. Xie, N. Vijakrishnan, and M. J. Irwin. Thermal-aware IP virtualization and placement for networks-on-chip architecture. In *Computer Design: VLSI in Computers and Processors. Proceedings. IEEE International Conference on*, pages 430–437, 2004.

- [HBB⁺11] J. Henkel, L. Bauer, J. Becker, O. Bringmann, U. Brinkschulte, S. Chakraborty, M. Engel, R. Ernst, H. Hartig, L. Hedrich, A. Herkersdorf, R. Kapitza, D. Lohmann, P. Marwedel, M. Platzner, W. Rosenstiel, U. Schlichtmann, O. Spinczyk, M. Tahoori, J. Teich, N. When, and H.-J. Wunderlich. Design and architectures for dependable embedded systems. In *Hardware/Software Codesign and System Synthesis, Proceedings of the 9th International Conference on*, pages 69–78, 2011.
- [HBK06] J. Held, J. Bautista, and S. Koehl. From a few cores to many: A tera-scale computing research review, 2006. Intel Research White Paper.
- [HHKS08] P. Hölzenpies, J. Hurink, J. Kuper, and G. Smit. Run-time spatial mapping of streaming applications to a heterogeneous multi-processor system-on-chip (MPSoC). In *Proceedings of the conference on Design, automation and test in Europe*, pages 212–217, 2008.
- [HSP] HSPICE User Guide: Simulation and analysis.
- [HZL⁺07] S. Hellebrand, C.G. Zoellin, S. Ludwig, T. Coym, and B. Straube. A refined electrical model for particle strikes and its impact on SEU prediction. In *Defect and Fault-Tolerance in VLSI Systems. 22nd IEEE International Symposium on*, pages 50–58, 2007.
- [Iee90] IEEE standard glossary of software engineering terminology, 1990. IEEE Std 610.12-1990.
- [Int04] Intel. Enhanced Intel Speedstep technology for the Intel Pentium M processor (intel white paper), 2004.
- [Int12] Intel Single-Chip Cloud Computer, abgerufen am 18.10.2012. <http://www.intel.com/content/www/us/en/research/intel-labs-singlechip-cloud-computer.html>.
- [Int13] Desktop 3rd generation intel core processor family and LGA1155 socket thermal mechanical specifications and design guidelines, abgerufen am 08.04.2013. <http://www.intel.com/content/www/us/en/processors/core/3rd-gen-core-lga1155-socket-guide.html>.
- [ITR12] International technology roadmap for semiconductors 2011 edition: 2012 update: Overall roadmap technology characteristics (ORTC) tables, abgerufen am 17.10.2012. <http://www.itrs.net/Links/2012ITRS/Home2012.htm>.
- [ITR13a] International technology roadmap for semiconductors 2007 edition: Design chapter, abgerufen am 16.08.2013. <http://www.itrs.net/Links/2007ITRS/Home2007.htm>.
- [ITR13b] International technology roadmap for semiconductors 2011 edition: 2012 update: Integration, devices, and structures (PIDS) tables, abgerufen am 17.01.2013. <http://www.itrs.net/Links/2012ITRS/Home2012.htm>.

- [Jan03] A. Jantsch. NoCs: a new contract between hardware and software. In *Proceedings of the Euromicro Symposium on Digital System Design*, pages 10–16, 2003.
- [Jed11] Failure mechanisms and models for semiconductor devices, 2011. JEDEC Publication JEP122G.
- [JGB01] S. L.-B. Javier, J. Garrido, and E. Boemo. Measurement of FPGA die temperature using run-time reconfiguration. In *in Proceedings of the 7th International Workshop on Thermal Investigations of ICs and Systems*, 2001.
- [Kal12] Kalray Multi-Purpose Processor Array MPPA 256, abgerufen am 11.10.2012. <http://www.kalray.eu/products/mppa-manycore/mppa-256/>.
- [KCBB06] E. Kursun, C.-Y. Cher, A. Buyuktosunoglu, and P. Bose. Investigating the effects of task scheduling on thermal behavior. In *In 3rd Workshop on Temperature-Aware Computer Systems*, pages 1–12, 2006.
- [KJS⁺02] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani. A network on chip architecture and design methodology. In *VLSI. Proceedings. IEEE Computer Society Annual Symposium on*, pages 105–112, 2002.
- [KKW03] G.A. Klutke, P.C. Kiessler, and M.A. Wortman. A critical look at the bathtub curve. *Reliability, IEEE Transactions on*, 52(1):125–129, March 2003.
- [KND02] F. Karim, A. Nguyen, and S. Dey. An interconnect architecture for networking systems on chips. *Micro, IEEE*, 22(5):36–45, September 2002.
- [Kru00] A. Krum. *Thermal Management in The CRC handbook of thermal engineering*. Springer, 2000.
- [KSS11] S. Kaushik, A. K. Singh, and T. Srikanthan. Preprocessing-based run-time mapping of applications on NoC-based MPSoCs. In *VLSI, IEEE Computer Society Annual Symposium on*, pages 337–338, 2011.
- [KTJR05] R. Kumar, D.M. Tullsen, N.P. Jouppi, and P. Ranganathan. Heterogeneous chip multiprocessors. *Computer*, 38(11):32–38, November 2005.
- [Lam12] J. Lamberson. Single and multistage watchdog timers, 2012. Sensoray White Paper.
- [LBS04] D. Lee, D. Blaauw, and D. Sylvester. Gate oxide leakage current analysis and reduction for VLSI circuits. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 12(2):155–166, February 2004.
- [LCN⁺09] W. Liu, A. Calimera, A. Nannarelli, E. Macii, and M. Poncino. On-chip thermal modeling based on SPICE simulation. In *Proceedings of the 19th international conference on Integrated Circuit and System Design*, pages 66–75, 2009.

- [LCOM08] H. G. Lee, N. Chang, U. Y. Ogras, and R. Marculescu. On-chip communication architecture exploration: A quantitative evaluation of point-to-point, bus, and network-on-chip approaches. *ACM Transactions on Design Automation of Electronic Systems*, 12(3):1–20, May 2008.
- [Lie06] J. Lienig. Introduction to electromigration-aware physical design. In *Proceedings of the international symposium on Physical design*, pages 39–46, 2006.
- [LRRB05] B. Le, T.W. Rondeau, J.H. Reed, and C.W. Bostian. Analog-to-digital converters. *IEEE Signal Processing Magazine*, 22(6):69–77, November 2005.
- [LWH⁺05] L.-Y. Lin, C.-Y. Wang, P.-J. Huang, C.-C. Chou, and J.-Y. Jou. Communication-driven task binding for multiprocessor with latency insensitive network-on-chip. In *Design Automation Conference. Proceedings of the ASP-DAC. Asia and South Pacific*, pages 39–44, 2005.
- [McC85] E. J. McCluskey. Built-in self-test techniques. *IEEE Des. Test*, 2(2):21–28, March 1985.
- [MCKW10] R. Manevich, I. Cidon, A. Kolodny, and I. Walter. Centralized adaptive routing for NoCs. *Computer Architecture Letters*, 9(2):57–60, February 2010.
- [MEB09] M. Monton, J. Engblom, and M. Burton. Checkpoint and restore for systemC models. In *Specification Design Languages. Forum on*, pages 1–6, 2009.
- [MFMB02] S.M. Martin, K. Flautner, T. Mudge, and D. Blaauw. Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads. In *Computer Aided Design. IEEE/ACM International Conference on*, pages 721–725, 2002.
- [Mil92] MIL-STD-1547, 1992. Military Standard, Electronic Parts, Materials, and Processes for Space and Launch Vehicles.
- [MKSZ05] S Mitra, T. Karnik, N. Seifert, and Ming Zhang. Logic soft errors in sub-65 nm technologies design and CAD challenges. In *Design Automation Conference. Proceedings. 42nd*, pages 2–4, 2005.
- [Moo65] G. E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8):114–117, 1965.
- [Nic05] M. Nicolaidis. Design for soft error mitigation. *Device and Materials Reliability, IEEE Transactions on*, 5(3):405–418, September 2005.
- [Nio13] Altera Nios II Embedded Processor, abgerufen am 31.01.2013. <http://www.altera.com/devices/processor/nios2/ni2-index.html>.
- [Nur04] J. Nurmi. *Interconnect-Centric Design for Advanced SOC and NOC*. Springer, 2004.
- [Ope13] Open source hardware IP-cores, abgerufen am 31.01.2013. <http://opencores.org/projects> (abgerufen am 31.01.2013).

- [Pas11] Implementierung eines Simulators für die effiziente Exploration von Hardware/Software-Anwendungen in einem Network-on-Chip, 2011. Masterarbeit Peter Passow.
- [PGF⁺06] P.P. Pande, A. Ganguly, B. Feero, B. Belzer, and C. Grecu. Design of low power reliable networks on chip through joint crosstalk avoidance and forward error correction coding. In *Defect and Fault Tolerance in VLSI Systems. 21st IEEE International Symposium on*, pages 466–476, 2006.
- [PGIS03] P.P. Pande, C. Grecu, A. Ivanov, and R. Saleh. Design of a switch for network on chip applications. In *Circuits and Systems. Proceedings of the International Symposium on*, pages 217–220, 2003.
- [Pow13] IBM PowerPC 405 CPU core product overview, abgerufen am 03.05.2013. <https://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/3D7489A3704570C0872571DD0065934E>.
- [RCN03] J. M. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital integrated circuits*. Prentice Hall, 2003.
- [RFFM03] M.P. Robinson, K. Fischer, I.D. Flintoft, and A.C. Marvin. A simple model of EMI-induced timing jitter in digital circuits, its statistical distribution and its effect on circuit performance. *Electromagnetic Compatibility, IEEE Transactions on*, 45(3):513–519, September 2003.
- [Ris05] L. Risch. Pushing CMOS beyond the roadmap. In *Solid-State Circuits Conference. Proceedings of the 31st European*, pages 63–68, 2005.
- [RIT07] P. Rantala, J. Isoaho, and H. Tenhunen. Novel agent-based management for fault-tolerance in network-on-chip. In *Digital System Design Architectures, Methods and Tools. 10th Euromicro Conference on*, pages 551–555, 2007.
- [RK09] J.A. Rivers and P. Kudva. Reliability challenges and system performance at the architecture level. *Design Test of Computers, IEEE*, 26(6):62–73, November 2009.
- [RMI04] K. K. Ryu and V.J. Mooney III. Automated bus generation for multiprocessor SoC design. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 23(11):1531–1549, November 2004.
- [Rob06] J. Robertson. High dielectric constant gate oxides for metal oxide Si transistors. *Reports on Progress in Physics*, 69(2):327–396, 2006.
- [RTZ98] J. Rajska, J. Tyszer, and N. Zacharia. Test data decompression for multiple scan designs with boundary scan. *IEEE Trans. Comput.*, 47(11):1188–1200, November 1998.
- [S.09] Kubisch S. *Architekturen für Ethernet-basierte Teilnehmerzugangsnetzwerke und deren Umsetzung in Hardware*. PhD thesis, Institute of Applied Microelectronics and Computer Engineering, University of Rostock, 2009.

- [SB03] D. K. Schroder and J. A. Babcock. Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing. *Journal of Applied Physics*, 94(1):1–18, February 2003.
- [SCS⁺10] H. Sämrow, C. Cornelius, J. Salzmann, A. Tockhorn, and D. Timmermann. Utilizing parallelism of TMR to enhance power efficiency of reliable ASIC designs. In *Computer Engineering and Systems, International Conference on*, pages 251–256, 2010.
- [SIC10] SoI Industry Consortium. Fully depleted SoI - designed for low power, 2010.
- [Sie91] D.P. Siemwiorek. Architecture of fault-tolerant computers: an historical perspective. *Proceedings of the IEEE*, 79(12):1710–1734, May 1991.
- [SJKS10] A. K. Singh, W. Jigang, A. Kumar, and T. Srikanthan. Run-time mapping of multiple communicating tasks on MPSoC platforms. *Procedia Computer Science*, 1(1):1019–1026, May 2010.
- [SKH08] E. Salminen, A. Kulmala, and T. D. Hämäläinen. Survey of Network-on-Chip proposals, 2008. OCP-IP White Paper.
- [SLN98] D. Singh, II Lakin, D.R., and P. Nigh. Binning for IC quality: experimental studies on the SEMATECH data. In *Defect and Fault Tolerance in VLSI Systems. Proceedings. IEEE International Symposium on*, pages 4–10, 1998.
- [SMS09] T. Sundström, B. Murmann, and C. Svensson. Power Dissipation Bounds for High-Speed Nyquist Analog-to-Digital Converters. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 56(3):509–518, March 2009.
- [Son12] Sonics SonicsGN Network-on-Chip, abgerufen am 11.10.2012. <http://www.sonicsinc.com/SonicsGN.htm>.
- [SPKJ04] L. Shang, L.-S. Peh, A. Kumar, and N. K. Jha. Thermal modeling, characterization and management of on-chip networks. In *Proceedings of the 37th annual IEEE/ACM International Symposium on Microarchitecture*, pages 67–78, 2004.
- [SRK11] H. Shah, A. Raabe, and A. Knoll. Priority division: A high-speed shared-memory bus arbitration with bounded latency. In *Design, Automation Test in Europe Conference Exhibition*, pages 1–4, 2011.
- [SSH⁺03] K. Skadron, M.R. Stan, Wei Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *Computer Architecture. Proceedings. 30th Annual International Symposium on*, pages 2–13, 2003.
- [Sta02] J. H. Stathis. Reliability limits for the gate insulator in CMOS technology. *IBM Journal of Research and Development*, 46(2.3):265–286, March 2002.
- [STN02] D. Siguenza-Tortosa and J. Nurmi. Proteo: a new approach to network-on-chip. In *International Conference on Communication Systems and Networks*, pages 1–5, 2002.

- [Sys] IEEE standard SystemC language reference manual.
- [Sys13] Accellera core SystemC language 2.2, abgerufen am 03.05.2013. <http://www.accellera.org/downloads/standards/systemc>.
- [TB10] M. Tehranipoor and K.M. Butler. Power supply noise: A survey on effects and research. *Design Test of Computers, IEEE*, 27(2):51–67, March 2010.
- [TCST10] A. Tockhorn, C. Cornelius, H. Sämrow, and D. Timmermann. Modeling temperature distribution in networks-on-chip using RC-circuits. In *Design and Diagnostics of Electronic Circuits and Systems, 13th IEEE International Symposium on*, pages 229–232, 2010.
- [TDB⁺06] X.-T. Tran, J. Durupt, F. Bertrand, V. Beroule, and C. Robach. A DFT architecture for asynchronous networks-on-chip. In *Test Symposium. 11th IEEE European*, pages 219–224, 2006.
- [Til12a] Tilera TILE64 Multicore Processor Family, abgerufen am 11.10.2012. <http://www.tilera.com/products/processors/TILE64>.
- [Til12b] Tilera iMesh Architecture, abgerufen am 11.10.2012. <http://www.tilera.com/technology>.
- [TLM13] Accellera SystemC Transaction-Level Modling library 2.0.1, abgerufen am 03.05.2013. <http://www.accellera.org/downloads/standards/systemc>.
- [VHL⁺05] S. Velusamy, Wei Huang, J. Lach, M. Stan, and K. Skadron. Monitoring temperature in FPGA based SoCs. In *Computer Design: VLSI in Computers and Processors. Proceedings. IEEE International Conference on*, pages 634–637, 2005.
- [VSE⁺00] E. M. Vogel, J. S. Suehle, M. D. Edelstein, B. Wang, Y. Chen, and J. B. Bernstein. Reliability limits for the gate insulator in CMOS technology. *Electron Devices, IEEE Transactions on*, 47(6):1183–1191, June 2000.
- [WCTT10] T. Wegner, C. Cornelius, A. Tockhorn, and D. Timmermann. Monitoring and control of temperature in networks-on-chip. In *6th Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, pages 184–192, 2010.
- [WGT12] T. Wegner, M. Gag, and D. Timmermann. Performance analysis of temperature management approaches in networks-on-chip. *IJERTCS*, 3(4):19–41, June 2012.
- [WL03] D. Wiklund and Dake Liu. SoCBUS: switched network on chip for hard real time embedded systems. In *Parallel and Distributed Processing Symposium. Proceedings. International*, pages 8–15, 2003.
- [WZPM02] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik. Orion: a power-performance simulator for interconnection networks. In *Microarchitecture. Proceedings. 35th Annual IEEE/ACM International Symposium on*, pages 294–305, 2002.

- [YBM04] T. T. Ye, L. Benini, and G. De Micheli. Packetization and routing analysis of on-chip multiprocessor networks. *Journal of Systems Architecture*, 50(2):81–104, January 2004.

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus den Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form weder veröffentlicht noch einer anderen Prüfungsbehörde vorgelegt.

Rostock, 02. Januar 2014

Wissenschaftliche Beiträge

Die vorliegende Arbeit und die darin enthaltenen Ergebnisse basieren zu großen Teilen auf den im Folgenden chronologisch aufgeführten wissenschaftlichen Publikationen.

- Wegner, T.: Erhöhung der Zuverlässigkeit in Networks-on-Chip: Konzepte und Integrationsmethoden, VDM Verlag Dr. Müller, ISBN:978-3639263589, Rostock, Deutschland, Juni 2010
- Gag, M.; Wegner, T.; Timmermann, D.: System Level Power Estimation of System-on-Chip Interconnects in Consideration of Transition Activity and Crosstalk, International Workshop on Power and Timing Modeling, Optimization and Simulation, S. 21-30, ISBN:978-3-642-17751-4, Grenoble, Frankreich, September 2010
- Wegner, T.; Cornelius, C.; Gag, M.; Tockhorn, A.; Uhrmacher, A.: Simulation of Thermal Behavior for Networks-on-Chip, 28th NORCHIP Conference, S. 1-4, ISBN: 978-1-4244-8971-8, DOI:10.1109/NORCHIP.2010.5669472, Tampere, Finnland, November 2010
- Wegner, T.; Cornelius, C.; Tockhorn, A.; Timmermann, D.: Monitoring and Control of Temperature in Networks-on-Chip, 6th Doctoral Workshop on Mathematical and Engineering Methods in Computer Science - Selected Papers, OpenAccess Series in Informatics, S. 124-131, ISBN:978-3-939897-22-4, Schloss Dagstuhl, Deutschland, März 2011
- Wegner, T.; Gag, M.; Timmermann, D.; Uhrmacher, A.: Reduction of Thermal Imbalances and Hot Spots in Networks-on-Chip Using Proactive Temperature Management, 5. GMM/GI/ITG- Fachtagung Zuverlässigkeit und Entwurf, S. 84-91, ISBN: 978-3-8007-3357-6, Hamburg, Deutschland, September 2011
- Wegner, T.; Gag, M.; Timmermann, D.: Impact of Proactive Temperature Management on Performance of Networks-on-Chip, International Symposium on System-on-Chip 2011, S. 116-121, ISBN: 978-1-4577-0670-7, Tampere, Finnland, November 2011
- Gag, M.; Wegner, T.; Waschki, A.; Timmermann, D.: Temperature and On-Chip Crosstalk Measurement using Ring Oscillators in FPGA, 15th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems, S. 201-204, ISBN: 978-1-4673-1185-4, Tallinn, Estland, April 2012

- Wegner, T.; Gag, M.; Timmermann, D.: Performance Analysis of Temperature Management Approaches in Networks-on-Chip, International Journal of Embedded and Real-Time Communication Systems, S. 19-41, EISSN: 1947-3184, Hershey, Pennsylvania, USA, Dezember 2012
- Gag, M.; Wegner, T.; Gorski, P.; Tockhorn, A.; Timmermann, D.: System level modeling of Networks-on-Chip for power estimation and design space exploration, 16. Workshop Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen, S. 25-34, ISBN: 978-3-86009-147-0, Rostock-Warnemünde, Deutschland, März 2013
- Wegner, T.; Gag, M.; Timmermann, D.: Efficiency Analysis of Approaches for Temperature Management and Task Mapping in Networks-on-Chip, Advancing Embedded Systems and Real-Time Communications with Emerging Technologies Book Series, S. 368-398, ISBN-13: 978-1466660342, IGI Global, Juni 2014

Thesen

1. Die Skalierung der Fertigungstechnologien erfordert die Verlagerung eines rein leistungsorientierten Entwurfs digitaler Schaltungen hin zu einem Ansatz, welcher darüber hinaus auch Zuverlässigkeit und Energieeffizienz berücksichtigt.
2. Die Temperatur nimmt bezüglich der Zuverlässigkeit eine besondere Stellung ein. Insbesondere Temperaturspitzen und -ungleichgewichte sind für thermischen Stress und daraus resultierende Fehlermechanismen verantwortlich.
3. Die technologische Entwicklung führt zu einer zunehmenden Diskrepanz zwischen theoretisch möglicher Komplexität hochintegrierter Systeme und realer Entwurfseffizienz.
4. Die Architektur der Networks-on-Chip (NoCs) ist mit den Eigenschaften einer guten Skalierbarkeit sowie einer ausgeprägten Modularität in der Lage die technologischen sowie architekturellen Problemstellungen zu überwinden. NoCs werden somit den steigenden Anforderungen moderner hochintegrierter Systeme besser gerecht und ersetzen daher zunehmend busbasierte Medien.
5. Selbst hochgradig vernetzte NoC-basierte Kommunikationsarchitekturen sind nicht in der Lage, alterungs- und verschleißbedingte Ausfälle einzelner Netzwerkkomponenten befriedigend zu kompensieren.
6. Schaltvorgänge in digitalen Schaltungen führen zur Umwandlung von elektrischer Energie in Wärme. Zwischen dem Auftreten von Schaltvorgängen und einer entsprechenden Wärmeausbreitung innerhalb der Schaltung besteht eine vergleichsweise große Verzögerung.
7. Das verzögerte Eintreten von Temperaturerhöhung und Wärmeausbreitung kann genutzt werden, um Vorhersagen bezüglich der zu erwartenden Temperaturlentwicklung zu treffen.
8. Das thermische Verhalten hochintegrierter Schaltungen lässt sich mithilfe äquivalenter RC-Elemente schnell und akkurat modellieren.

9. Die Kombination des RC-basierten Temperaturmodells mit der funktionalen Simulation NoC-basierter Systeme ermöglicht eine laufzeitfähige Modellierung der Temperaturrentwicklung solcher Systeme.
10. Die Modellierung der Temperaturrentwicklung zur Laufzeit NoC-basierter Systeme bildet im Zusammenhang mit der Verzögerung zwischen Schaltvorgängen und daraus hervorgehenden Temperaturänderungen die Grundlage für ein temperaturorientiertes proaktives Systemmanagement.
11. Unabhängig von der eingesetzten Strategie für das Temperaturmanagement NoC-basierter Systeme ist stets ein Kompromiss zwischen der erzielten Verbesserung des thermischen Verhaltens und der damit einhergehenden Einbußen bezüglich der Leistungsfähigkeit erforderlich.
12. Die durch ein Temperaturmanagement verursachten Leistungsdefizite lassen sich mithilfe eines proaktiven Ansatzes reduzieren. Der positive Einfluss auf die Temperaturverteilung kann im Vergleich zu etablierten Konzepten deutlich besser erhalten werden.
13. Ein Task-Mapping zur Laufzeit NoC-basierter Systeme ist für dynamisch wechselnde Anwendungsprofile besser geeignet, als ein statisches Mapping während der Entwurfsphase.
14. Der Vorteil eines dynamischen Task-Mappings zur Laufzeit basiert auf der Einbeziehung der Verfügbarkeit der Systemressourcen und des Zustands der Kommunikationsinfrastruktur in die Mapping-Entscheidungen.
15. Ein proaktives temperaturorientiertes Task-Mapping ist gegenüber etablierten distanz- oder lastorientierten Ansätzen in der Lage das Temperaturprofil NoC-basierter Systeme auszubalancieren, ohne Einbußen bezüglich der Leistungsfähigkeit zu verursachen.

Kurzreferat

Die Leistungsfähigkeit nanoelektronischer Systeme wird bislang durch die Skalierung der Halbleitertechnologien kontinuierlich gesteigert. Aktuelle Technologien stoßen jedoch vermehrt an physikalische Grenzen. Neben der Leistungsfähigkeit gewinnt damit die Zuverlässigkeit zunehmend an Relevanz. Dabei sind besonders thermische Aspekte von Bedeutung, da diese Alterungs- und Verschleißerscheinungen beeinflussen. Darüber hinaus bewirkt die enorme Komplexität hochintegrierter Systeme einen Paradigmenwechsel bezüglich des Systementwurfs hin zu kommunikationsorientierten Ansätzen. Networks-on-Chip (NoCs) sind in der Lage sowohl die technologischen als auch die architekturellen Probleme zu überwinden. Diese Arbeit leistet diesbezüglich einen wertvollen Beitrag, um die Tendenzen des zuverlässigkeitsorientierten Systementwurfs und -betriebs sowie die Nutzung NoC-basierter Konzepte zu vereinen.

Zunächst wird die technologische Entwicklung digitaler Schaltungen betrachtet. Besonderes Augenmerk wird dabei auf die Zuverlässigkeit und Lebensdauer hochintegrierter Schaltungen gelegt. Darüber hinaus wird die Signifikanz des Einflusses thermischer Aspekte auf Ausfall- und Verschleißerscheinungen verdeutlicht. Weiterhin wird ein Überblick über die Kommunikationsstrukturen nanoelektronischer Systeme gegeben. Neben den herausgearbeiteten Vorteilen NoC-basierter Systeme werden auch die Auswirkungen des Ausfalls von Netzwerkkomponenten hervorgehoben, welche selbst durch hochgradig vernetzte NoC-Strukturen nicht in befriedigendem Maße kompensiert werden. Somit gilt es, den häufig temperaturbedingten Ausfällen vorzubeugen. Das zu diesem Zweck angestrebte Konzept einer proaktiven Beeinflussung der Temperaturentwicklung basiert auf der Verzögerung zwischen Schaltvorgängen digitaler Schaltungen und einer entsprechenden Temperaturerhöhung. Dass diese Verzögerung ausreichend groß ist, um Vorhersagen treffen zu können, wird anhand eines FPGA-basierten Versuchsaufbaus belegt.

Im weiteren Verlauf wird ein Temperaturmodell erarbeitet, welches für das angestrebte proaktive Konzept erforderlich ist. Dieses Modell bildet das thermische Verhalten hochintegrierter Schaltkreise mithilfe äquivalenter RC-Elemente nach. Durch die Kombination mit einer funktionalen Simulation NoC-basierter Systeme wird damit eine laufzeitfähige Modellierung der Temperaturentwicklung ermöglicht. Des Weiteren wird nachgewiesen, dass sowohl Modellierungsgenauigkeit als auch -geschwindigkeit den Anforderungen für ein proaktives Systemmanagement genügen.

Die praktische Anwendbarkeit des Temperaturmodells wird mithilfe zweier Anwendungsszenarien belegt. Dies betrifft ein proaktives Temperaturmanagement sowie ein proaktives Task-Mapping zur Laufzeit NoC-basierter Systeme. Beide Konzepte machen sich den Vorteil einer vorhersagbaren Temperaturentwicklung zu nutze, um entweder die Leistungsfähigkeit dynamisch anzupassen oder Tasks zu platzieren. Für den Nachweis der praktischen Relevanz solcher proaktiver Konzepte werden jeweils etablierte Vergleichskonzepte herangezogen.

Abstract

Scaling of semiconductor technology leads to a continuous growth in performance of nanoelectronic systems. However, current technologies reach their physical limits. As a result reliability becomes a key aspect for the design and the operation of complex systems. Especially thermal issues gain in importance, since they influence effects concerning aging and wearout. Furthermore, the enormous complexity of highly integrated systems causes a paradigm shift of system design towards communication centric concepts. Regarding this, Networks-on-Chip (NoCs) are able to overcome both the technological as well as the architectural issues. This work provides valuable contributions in order to combine reliability aware system design and operation with the concept of NoCs.

Initially, the technological development of digital circuits is examined focusing on reliability and lifetime. In addition, the significant influence of thermal aspects on deterioration and failures is highlighted. Furthermore, an overview over the communication structures of modern nanoelectronic systems is given. Besides the advantages of NoC-based systems the severe impact of network failures is distinguished emphasizing the inability of NoC-based structures to compensate such failures. Consequently, failure mechanisms often related to temperature have to be precluded. For this purpose, a concept proactively influencing the temperature development is aspired. An FPGA-based experimental setup serves to prove the feasibility of this approach.

In the course of this work a temperature model is developed, which is required for the aspired concept of proactive system management. This model emulates the thermal behavior of highly integrated circuits using equivalent RC elements. By combining the temperature model with a functional simulation of NoC-based systems, modeling of temperature development during system runtime is facilitated. Furthermore, it is shown that accuracy and performance of temperature modeling meet the requirements of a proactive system management.

The practicability of the temperature model is demonstrated using two application scenarios. Namely, these are runtime approaches for a proactive temperature management and a proactive task mapping. Both concepts exploit the possibility of predicting future temperature developments in order to dynamically scale system performance or map tasks, respectively. To verify the practical relevance of these proactive approaches, commonly used concepts are consulted.