

A Highly Dependable, Analog Multi-Core Mixed-Signal Task Distribution System

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften

vorgelegt beim Fachbereich Informatik und Mathematik
der Goethe-Universität
in Frankfurt am Main

von
Julius von Rosen
aus Frankfurt

Frankfurt (2014)
(D 30)

vom Fachbereich Informatik und Mathematik
der Goethe-Universität als Dissertation angenommen

Dekan: Prof. Dr. Uwe Brinkschulte

Gutachter: Prof. Dr. Lars Hedrich
Prof. Dr. Uwe Brinkschulte

Datum der Disputation: 9. Juni 2015

Acknowledgments

Abstract

The objective of this thesis is to develop an analog architecture, which distributes decentralized, highly dependable and self-reliant tasks within a mixed-signal multi-core System-on-Chip. Hence, every step of the design process and their results to develop such an analog system are presented, since the synthesis of analog circuits is still mostly done manually, contrary to the almost fully automatized and formalized design of digital circuits. Especially the validation and long-term satisfaction of the specification of the analog components need a high degree of verification and testing done by hand.

The usability of the design is evaluated against known task distribution approaches, which are either highly sophisticated digital circuits or software implementations, which have shown their usability in real-world applications already. Further, evaluations are done against analog approaches, which are capable of being transformed to match the objectives of this thesis. This allows to clearly differentiate the developed and designed architecture from the existing approaches.

The design of an artificial, analog hormone system is a bio-inspired replica to distribute information and tasks within a system. The endogenous transmitters are mapped to voltages and currents, which are, properly coordinated, spreading throughout the full system. Those hormones can be applied locally, but are noticed globally at every core. However, the physical laws of electrical engineering have to be taken into consideration as equally as the balance of the hormones to guarantee the reliable and dependable functionality of the task distribution.

Within the design process a complete formal description of the analog components of the hormone system is done. Based on the description, solving the differential equations and inequalities, which mirror the behavior of the hormone system, enables to issue a reliability analysis. With this analysis the fail-safety of the components are classified, the dependencies of the circuit parameters are determined and a set of specification, needed for the design process, are derived. Further, a robustness value has been defined, which quantifies the interval of

legitimacy within which any process variation, noise and similar effect is fully absorbed - the correct functionality of the hormone system is unharmed.

Next to the specification of the components, the prototypical implementation of the system is done, followed by the placing, routing and layouting of the components. Lastly, the layout has been fabricated. The results of the simulations of the implementations, the extracted view and the measurements of the prototypical chip are presented and compared to evaluate the analog hormone system against the presented approaches. The detailed comparison classifies each presented approach and the hormone system based on the following key points, which are essential for autonomous task distribution architectures:

- self-control,
- size increase,
- real-time capability,
- reliability gain,
- scalability,
- mixed-signal capability.

Additionally, monitor circuits are designed, which also increase the reliability of the hormone system and the analog cores. The monitor circuits enable a self-reliant management of the cores by themselves. In turn, the self-reliance guarantees the self-control of each core, distributing self-optimized and decentralized tasks within the system. The focus is on monitoring general performance changes, instead of single failure effects like *electromigration* or *Hot Carrier Injection*. If the changing performances remain within the reliability interval, the hormone system has no need to react to, respectively counteract, the change of the system behavior. If the change exceeds the reliability interval though, the monitors are issuing task reallocations.

Concluding, the thesis presents the complete design process of a reliable architecture, which distributes tasks within a mixed-signal multi-core System-on-Chip highly dependable, decentralized and self-reliant. The design process ends with the fabrication of a prototypical chip. Yet, to truly state the self-control property, further monitor circuits for autonomous operating robots are indispensable. Also, extensive studies concerning the real-time capability of the hormone system within different fields of application can only be carried out, if the prototypical implementation of the analog hormone system is attached to designed analog cores, fabricated and measured.

Zusammenfassung (German Abstract)

Gegenstand dieser Dissertation ist die Entwicklung einer analogen Architektur, die dezentral, hoch-verlässlich und selbstständig Aufgaben innerhalb eines Mixed-Signal Mehrkern-System-on-Chips verteilt. Dazu werden im Rahmen dieser Arbeit auch alle entscheidenden Zwischenschritte und deren Ergebnisse, die während des Entwurfsprozesses für das Design der analogen Schaltungen entscheidend sind, vorgestellt.

Im Gegensatz zum Entwurfsprozess digitaler Schaltungen, die weitestgehend automatisiert und formalisiert sind, bedarf es bei der Synthese von analogen Schaltungen fast ausschließlich manueller Schritte. Insbesondere im Bereich der Validierung und dauerhaften Sicherstellung der spezifizierten Eigenschaften der analogen Schaltungen, ist eine manuelle Verifikation erforderlich.

Gemessen wird die Nutzbarkeit des Entwurfs anhand der in der Praxis bisher gängigen Verfahren zur zuverlässigen Aufgabenverteilung. Diese beziehen sich jedoch ausschließlich auf Software-Implementierungen, verteilt auf Prozessorkernen, oder hochentwickelte, digitale Schaltungen. Des Weiteren werden aus dem analogen Anwendungsbereich zwei Verfahren herangezogen, die auch auf die Thematik dieser Dissertation übertragen werden können. Für jedes dieser vorgestellten Verfahren wird eine Charakterisierung vorgenommen, um die Verfahren gegeneinander zu bewerten. Dadurch lässt sich die entwickelte Architektur klar von den bestehenden Verfahren abgrenzen, da ein solches analoges System zur Verteilung von Aufgaben bisher nicht existiert. Die herangezogenen Verfahren, die als Aufgabenverteilungs-Architektur realisiert werden können, sind

- ein Agenten-basiertes System,
- ein künstliches neuronales Netzwerk,
- ein analoges Voting Verfahren, sowie
- ein digitales, künstliches Hormon System.

Für den umfassenden Vergleich wird das Agenten-basierte Verfahren als Orientierung genutzt, anhand dessen geprüft wird, wie die anderen Verfahren bestehen. Damit lässt sich eine Rangliste der Verfahren anfertigen.

Der Entwurf eines künstlichen, analogen Hormonsystems (AAHS) ist die Abbildung eines Systems aus der Natur zur Verteilung von Informationen und Aufgaben. Hormone werden abstrahiert als Ströme und Spannungen dargestellt. Diese müssen jedoch entsprechend koordiniert über Leitungen verteilt werden. Die Auswertung und Steuerung der Hormone geschieht lokal in den Entscheidungseinheiten, die jeweils an einem Kern angeschlossen werden. Damit stellen die Entscheidungseinheiten dezentrale und redundante Bauteile dar. Allerdings müssen die Hormone auch global allen anderen Kernen zugänglich gemacht werden, damit diese ihre Entscheidungen auf gleiche beziehungsweise ähnliche Hormonwerte stützen. Ansonsten werden unter Umständen fehlerhafte Allokationen durchgeführt. Dies kann im besten Fall eine nicht optimale Aufgabenverteilung zur Folge haben, jedoch im schlechtesten Fall zu Doppel-Allokationen oder ähnlichem führen. Daher ist die Einhaltung bestimmter zeitlicher Schranken von großer Bedeutung. Auch müssen die physikalischen Gesetze der Elektrotechnik weiterhin berücksichtigt werden, damit das Gleichgewicht der Hormone gewahrt wird, um die Funktionalität der zuverlässigen Aufgabenverteilung zu gewährleisten.

Im Rahmen des Entwurfsprozesses erfolgt eine vollständige formale Beschreibung der analogen Komponenten des Hormonsystems. Auf Basis dieser formalen Beschreibung wird ein Differentialgleichungssystem aufgestellt. Dieses muss um Ungleichungen erweitert werden, um das gesamte Schaltungsverhalten abzubilden. Dazu muss das kontinuierliche System zu unterschiedlichen Zeitpunkten diskretisiert werden. Die Lösung dieses diskreten Systems ermöglicht nicht nur eine Zuverlässigkeitsanalyse, sondern auch die Bestimmung der verschiedenen Abhängigkeiten der Hormone und die Spezifikation der Komponenten, die für den Entwurf benötigt werden.

Die Zuverlässigkeitsanalyse klassifiziert jedes einzelne Bauteil anhand seiner Ausfallsicherheit. Durch die Analyse kann eine Aussage getroffen werden, wie wahrscheinlich ein Fehler in den verschiedenen Fehlerklassen auftritt, um dadurch unter anderem auch Rückschlüsse auf die Zuverlässigkeit der Aufgabenverteilung und des Gesamtsystems ermöglichen. Die Abhängigkeit der verschiedenen Bauteile, respektive der Hormonwerte ist ebenso wichtig für den Entwurfsprozess, wie die Spezifikation der einzelnen Komponenten, die sich anhand der Zuverlässigkeitsanalyse definieren lassen. Für die Spezifikation werden ver-

schiedene Kriterien herangezogen, unter anderem die genutzte Technologie, damit verbunden die Größe der Bauteile und die Versorgungsspannung, und die Verwendung des Bauteils innerhalb der Hormonarchitektur.

Auch wird ein Robustheitsfaktor definiert. Dieser bestimmt ein Zulässigkeitsintervall, innerhalb dessen jegliche Prozessschwankungen, aber auch Rauschen und ähnliche Störverhalten, abgefangen werden, ohne die korrekte Funktionalität des Hormonsystems zu beeinflussen. Somit werden für das analoge, künstliche Hormonsystem drei zuverlässigkeitsrelevante Bewertungen definiert:

1. die Ausfallsicherheit $P(X = \Psi) : \Psi \in \text{Fehlerklasse } \{A, B, C\}$,
2. die Kern-Ausfall-Toleranz F_t und
3. der Robustheitsfaktor r_{CC} .

Die Kern-Ausfall-Toleranz betrachtet das Gesamtsystem und wie sich der Verlust von Kernen im ungünstigsten Fall auf das Gesamtsystem auswirkt. Je höher der Wert, desto länger kann das System ausfallende Kerne kompensieren. Im Normalfall liegt der Wert F_t zwischen (0..1), jedoch existiert ein Sonderfall für den $F_t = 0$ definiert werden muss. Dieser Sonderfall inkludiert all die Fehlerzenarien, in denen ein einziger auftauchender Fehler zum Verlust des Gesamtsystems führt. Mit den drei definierten zuverlässigkeitsrelevanten Bewertungen kann die Zuverlässigkeit genau bestimmt und damit auch die Verbesserung der Zuverlässigkeit beziffert werden.

Dem Entwurfsprozess folgt nach der Spezifikation der Komponenten die prototypische Implementierung der Schaltungen. Für diese wurde auf ein semi-automatisiertes Synthese-Framework zurückgegriffen, die den sehr zeitaufwändigen Prozess des Entwurfes der Operationsverstärker signifikant verkürzt. Mit dem Synthese-Framework konnten die sechs Operationsverstärker für die zwei Hormonarchitekturen implementiert werden:

- der lokale Addierer, ein lokaler Schmitt-Trigger und der globaler Addierer,
- zwei lokale und ein globaler Transkonduktanzverstärker.

Jeder dieser Operationsverstärker erfüllt die teilweise sehr anspruchsvollen Spezifikationen, wie beispielsweise den geringen Overshoot bei den Operationsverstärkern oder die geringe R_{Last} bei einem der Transkonduktanzverstärkern. Zusätzlich sind zwei weitere Schmitt-Trigger entworfen worden, um den unterschiedlichen Anforderungen der Entscheidungseinheiten gerecht zu werden. Die Schmitt-Trigger basieren auf Sechs-Transistor Modellen.

Nach der Implementierung folgt das Layouten inklusive der Platzierung und Verdrahtung der einzelnen Komponenten. Die Simulationsergebnisse der Implementierungen lassen sich anschließend mit den Simulationsergebnissen der extrahierten Sicht vergleichen und analysieren. Veränderungen des Schaltungsverhaltens auf Grund der Fertigung werden erstmals sichtbar, zeigen jedoch, dass das analoge, künstliche Hormonsystem die erwartete Zuverlässigkeit erfüllt und nachweislich funktioniert. Anschließend wurde das Layout lithografiert und gefertigt, so dass neben den ersten Simulationsergebnissen, auch das analoge Hormonsystem auf einem gefertigten Chip ausgemessen werden kann. Die Ergebnisse dieser drei Auswertungen werden präsentiert und verglichen, um das künstliche, dezentrale, analoge Hormonsystem entsprechend gegen die anderen Verfahren zu bewerten. Ein umfassender Vergleich ermöglicht eine eindeutige Klassifizierung der unterschiedlichen Ansätze. Dabei liegt der Fokus auf folgenden Punkten, die im Rahmen der Anwendungsbereiche von autonomen Aufgabenverteilungsarchitekturen von Bedeutung sind:

- Selbst-Kontrolle,
- Flächenzuwachs,
- Echtzeitfähigkeit,
- Zuverlässigkeitssteigerung,
- Skalierbarkeit,
- Mixed-Signal Fähigkeit.

Auch werden die Vor- und Nachteile der Strom- und Spannungsbasierten Architekturen aufgezeigt, so dass eine Präferenz abgegeben werden kann, welche die zu bevorzugende AAHS Implementierung ist und welchen Kompromiss diese dafür verlangt.

Neben der Hormonarchitektur sind auch Monitor-Schaltungen entworfen worden, die die Zuverlässigkeit des Hormonsystems, sowie der analogen Kerne, erhöhen. Die Monitore ermöglichen die Selbstständigkeit der Kerne und erfüllen damit eine weitere Bedingung der Selbstkontrolle, um als autonome Systeme agieren zu können. Des weiteren gilt, dass eine verlässliche Selbstkontrolle die selbst-optimierende und dezentrale Aufgabenverteilung im Gesamtsystem gewährleistet. Dabei liegt der Fokus nicht auf einzelnen Fehlereffekten wie *Electromigration* oder *Hot Carrier Injection*, sondern auf der allgemeinen Veränderung der Schaltungen, wie dies zum Beispiel mit dem Spannungsdrift-Monitor geprüft

wird. Bewegt sich die Veränderung innerhalb des Zulässigkeitsintervalls, bedarf es praktisch keinem Eingreifen. Veränderungen aber, die das Intervall verletzen, führen Reaktionen bei der Aufgabenverteilung nach sich. Der Spannungsdrift-Monitor misst die Differenz an den Eingangstransistoren und lässt diese gegen die Schwellspannung eines Schmitt Triggers laufen. Wird die Schwellspannung überschritten, dann zeigt der Trigger *Verstärker defekt* an. Sinkt nun die Eingangsspannung wieder, so dass die negative Schwellspannung des Triggers unterschritten wird, dann liegt am Ausgang des Schmitt Triggers wieder die Versorgungsspannung an, der Monitor klassifiziert den Kern wieder als *gesund und aktiv*.

Auch wenn in dieser Arbeit ein produzierter Chip vorgestellt wird, der nachweisbar die richtige und verlässliche Funktionsweise zeigt, an Hand dessen Aufgaben zuverlässig auf einem Mixed-Signal Mehrkern-System-on-Chip verteilt werden, braucht es zum einen weitere Monitor-Schaltungen um der Eigenschaft der Selbst-Kontrolle, die für autonom agierende Roboter unabdingbar ist, vollständig gerecht zu werden und zum anderen ein Re-Design, um alle Single-Point-of-Failure (auch Leitungsverluste) zu eliminieren. In der Arbeit wurden einige Lösungsvorschläge zum Minimieren der Single-Point-of-Failure unterbreitet. Interessant wäre es, diese vollständig zu implementieren und zu fertigen. Dies würde eine Bewertung ermöglichen, in der eine weitere Minimierung der Ausfallsicherheit gegen die zusätzlich benötigte Fläche durchgeführt wird. Auch ließe sich die zeitliche Veränderung der Systeme zeigen.

Zudem müssten prototypische Chips mit den Aufgaben-ausführenden Kernen in Verbindung mit dem Hormonsystem entworfen und produziert werden, um weitere Studien zur echtzeitfähigen Einsetzbarkeit des Hormonsystems in unterschiedlichen Anwendungsbereichen durchführen zu können. Diese prototypischen Gesamtsysteme könnten wiederum auch genutzt werden, um die Veränderungen im Verhalten des Hormonsystems zu zeigen, wenn der Chip Umwelteffekten, wie Strahlungen und/oder Hitze, ausgesetzt ist.

Contents

Acknowledgments	i
Abstract	iii
Zusammenfassung (German Abstract)	v
List of Tables	xvi
List of Figures	xix
List of Symbols and Abbreviations	xxi
1 Introduction	1
1.1 Analog Circuit Design Flow	2
1.2 Circuit Reliability	4
1.3 Reliability-Aware Architectures	15
1.3.1 Artificial Neural Networks	17
1.3.2 Analog Voting	22
1.3.3 Artificial Hormone System	26
1.4 Motivating Example	29
1.5 Publications	30
1.6 Overview	31
2 Comparison of Reliable Architectures	33
2.1 Size Overhead	36
2.2 Reliability Gain	42
2.3 Real-Time Bounds	45
2.4 Summary of the Comparison	48
2.5 Contributions	51

3	Reliable, Mixed-Signal Architecture	53
3.1	Artificial Hormone System with Analog Components	54
3.2	Analog Artificial Hormone System	55
3.2.1	Defining the Analog Hormone System	59
3.2.2	Designing the Analog Hormone System	64
3.2.2.1	The Decision Module	65
3.2.2.2	The Hormone Bus	68
3.2.2.3	Comparing the Architectures	72
3.2.2.4	Open Design Measures	72
3.2.3	Task (Re-)Allocation Process	73
3.2.3.1	Allocating Tasks	74
3.2.3.2	Migrating within one Domain	76
3.2.3.3	Migrating between Domains	77
4	Dependability Analysis	79
4.1	Design Analysis	80
4.1.1	Algebraic Description	82
4.1.2	Symbolic Solution	85
4.2	Reliability Analysis	91
4.2.1	Failure Classification	92
4.2.2	Fail-Safety Investigation	94
4.2.3	Failure Sensitivity	100
4.2.4	Monitor Decisions	104
5	Design Methodology	105
5.1	Specification Generation	106
5.2	Semi-Automated Analog Circuit Design	109
5.3	Hormone System Design	112
5.3.1	Hormone System Synthesis	113
5.3.2	Hormone System Layout	118
5.3.3	Full System Task Migration	122
5.4	Monitoring	123
5.4.1	Hormone System Monitoring	123
5.4.2	Working Core Monitoring	125
5.4.3	Hormone Bus Monitoring	128

6	Results	129
6.1	Validating the Design	130
6.1.1	Simulation Runs of the Architectures	133
6.1.2	Simulation Runs of the Extracted View of the Architectures	141
6.1.3	Hardware Measurements	147
6.2	Meet the Challenge	149
6.3	Application Usage	155
7	Conclusions	159
7.1	Summary	159
7.2	Challenges and Future Work	161
A	Appendix	165
A.1	Major Functions of the Algebraic Analysis	165
A.2	Determine the Number of Voters	167
A.3	Layouts	168
	Bibliography	173

List of Tables

1.1	Voter Costs	25
2.1	Size Increase of the Different Reliable Architectures	36
2.2	Comparison of the Failing Core Tolerance	45
2.3	Summary of the Comparison Results	50
3.1	Size Increase of the Analog Artificial Hormone System	61
3.2	Preliminary Summary of the Analog Hormone System	62
4.1	Evaluation of the Symbolic Analysis	86
4.2	Evaluation of Combined Sets	88
4.3	Evaluation of the Specific Set	88
4.4	Evaluation Comparison of $\theta_{\gamma,i}$ and E_{γ}	88
4.5	Evaluation of the Corner Case Analysis	90
4.6	Failure Class Occurrences	93
4.7	Failure Susceptibility	101
4.8	Failure Sensitivity of Sample Working Cores	103
5.1	Derived Hormone Values	107
5.2	Generating the Specification	110
5.3	Set of Derived Specifications	111
5.4	Measurement Table of the Semi-Automated Synthesized	115
6.1	Classification of the System States	130
6.2	Signals of the Voltage-Based Architecture	135
6.3	Signals of the Current-Based Architecture	139
6.4	Timing Constraints of the Current-Based Architecture	139
6.5	Signals of the Extracted View of the Voltage-Based Architecture	141
6.6	Timing Constraints of the Simulation of the Extracted View	142

- 6.7 Hysteresis of the Schmitt Trigger of the Voltage-Based Architecture 143
- 6.8 Signals of the Extracted View of the Current-Based Architecture . . 145
- 6.9 Timing Constraints of the Simulation of the Extracted View 146
- 6.10 Hysteresis of the Schmitt Trigger of the Current-Based Architecture 147
- 6.11 Comparing the Timing Constraints 150
- 6.12 Comparing the Failure Class Areas 151
- 6.13 Measured Eager Value Allocation Bounds 151
- 6.14 Advantages of the Architectures 152
- 6.15 Completed Summary of AAHS 153
- 6.16 Real Comparison of AHS and AAHS 154

- A.1 Determine the Minimum Number of Voters 169

List of Figures

1.1	Y-chart of the Analog Design	3
1.2	Failure Severity Diagram	5
1.3	Performance Degradation of Different Reliability-Aware Approaches	10
1.4	Sine Input Sigal with Resulting Probability for Defect	11
1.5	Analog, Adaptive Body Biasing Based NBTI Monitor	12
1.6	On-Chip Monitor to Detect HCI and NBTI Degradation	13
1.7	Centralized Reliability-Aware Architectures	16
1.8	Reliable Architecture using ANN	18
1.9	Artificial Neuron	19
1.10	Implementation of an Artificial Neural Network	21
1.11	Artificial Neural Networks as Task Distribution System	23
1.12	Reliable Architecture using AV	24
1.13	Reliable Architecture using AHS	27
1.14	Assignment of Generalized Tasks	27
1.15	Artificial Hormone Loop	28
1.16	Failing of a Motor Control using a PID-Controller	29
1.17	Failing of a Signal Filtering	30
2.1	Different Reliability-Aware Architectures	35
2.2	Allowed Overhead of ANN and AHS	41
3.1	Digital Hormone Loop for Analog Cores	55
3.2	ABS Braking System using the AHS/AHS-A Architecture	56
3.3	Model of a Multi-Core System	57
3.4	Analog Artificial Hormone Based Control Loop	59
3.5	Sketches outlining $\tau_{G,i}$ and $\tau_{stable,i}$	60
3.6	Analog Hormone Loop for Digital Cores	63
3.7	Implementation of the Decision Module with OpAmps	65

3.8	Implementation of the Decision Module with OTAs	67
3.9	Voltage-based Hormone Bus Structure	70
3.10	Current-based Hormone Bus Structure	71
3.11	Communication Logic of a Core	75
3.12	Analog State Transfer	76
3.13	Task Migration between an Analog and a Digital Core	77
4.1	Sketch of the Analog Hormone System	80
4.2	Block Diagram of a Limiting Adder Circuit	81
4.3	Four Polyhedrons Representing Different Feasible Regions	89
4.4	Chebyshev Sphere inside the Polyhedron	90
4.5	Polyhedron representing Feasible Region of the Corner Case	90
4.6	Degradation Effects of a Schmitt Trigger	92
4.7	Classifying the Fail-Safety of the Decision Modules	94
4.8	Sensitivity Diagram of the Analog Hormone System	102
5.1	Flow of the Design Methodology of the Hormone System	106
5.2	Fully Automated Analog Synthesis Framework Flow	112
5.3	Schematics of the Voltage Adder	114
5.4	Schematics of the OTAs and ST	116
5.5	Schematics of the Decision Modules using OpAmps	117
5.6	Schematic of the Decision Module using OTAs	118
5.7	Schematics of the Architectures	119
5.8	The Fully Laidout Architectures	121
5.9	Voltage Drift Monitor	124
5.10	Block Diagram of a Monitor Circuit for an Output Stage	126
5.11	Monitoring the Supply Voltage of a Battery	127
5.12	Heartbeat Signal Monitor	128
6.1	Simulation Run of the Voltage-Based Architecture	134
6.2	Allocation Processes of the Voltage-Based Architecture	137
6.3	Simulation Run of the Current-Based Architecture	138
6.4	Visualized Timing Behavior of the Current-Based Architecture	140
6.5	Extracted View Simulation of the Voltage-Based Architecture	142
6.6	Re-Issued Simulation of the Extracted View	144
6.7	Simulation Run of the Current-Based Architecture	146
6.8	Photograph of the Test Chip	148
6.9	Reliability Prove of the Current-Based Architecture	148

6.10	Motor Control using AAHS	155
6.11	Simulation Result of the <i>Right Arm</i> Motor Control	156
6.12	Simulation Result of the Signal Filtering	157
6.13	Signal Filtering using AAHS	158
A.1	Layout of the Decision Modules of the Architectures	170
A.2	Layout of the Global Units of the Architectures	171

List of Symbols and Abbreviations

Notation

ω	Word containing a set of letters
Ω	Set of words
$ x $	Absolute value of x
$\dot{\mathbf{x}}$	Temporal derivative of $\mathbf{x}(t)$
$1..x$	Sequence of an ordered list of integers from 1 to x

Symbols

τ_X	Time constant of component X
N	Number of cores
m	Number of tasks
SR_E	Slew Rate of the eager value
O	Size Overhead
\mathbf{t}, \mathbf{f}	State of a core having a task allocated
α, β	Scalar factors
θ_X	Trigger value of component X
ϵ	Error function
T_i	Task i
C_γ	Core γ
$A_{\gamma,i}$	Local Accelerator Hormone
$E_\gamma, E_{\gamma,i}$	Eager Value Hormone
G_i	Global Hormone Level

List of Symbols and Abbreviations

$H_{\gamma,i}$	Local Hormone Level
S_i	Global Suppressor Hormone
F_t	Failing Core Tolerance
r_{CC}	Robustness Value
\square_X	Area of X
χ	Component of AAHS
C_0	State Coverage
C_1	Allocation Coverage
\mathcal{O}	Asymptotic upper complexity bound
H	Frequency Response

Abbreviations

AC	Alternating Current
AAHS	Analog Artificial Hormone System
AHS	Artificial Hormone System
AHS-A	Artificial Hormone System with Analog Cores
AMAS	Auction-based Multi-Agent System
AMS	Analog/Mixed-Signal
ANN	Artificial Neural Networks
BTI	Bias Temperature Instability
CAD	Computer-Aided Design
CM	Current Mirror
CMOS	Complementary Metal Oxide Semiconductor
DAE	Differential Algebraic Equation
DC	Direct Current
DF	Design Failures
DRC	Design Rule Check
EDA	Electronic Design Automation
EE	Environmental Effects
EM	Electromigration

FPAA	Field Programmable Analog Arrays
FPGA	Field-Programmable Gate Array
HDL	Hardware Description Language
HCI	Hot Carrier Injection
HNN	Hardware Neural Networks
IVR	Input Voltage Range
KCL	Kirchhoff's Current Law
KVL	Kirchhoff's Voltage Law
LVS	Layout Versus Schematic
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
MTDC	Multiple Task Distribution Controllers
NRD	Non-Recoverable Degradation
NMR	<i>N</i> -tuple Modular Redundancy
NP	Nondeterministic Polynomial Time
ODE	Ordinary Differential Equation
OpAmp	Operational Amplifier
OR	Output Resistance
OTA	Operational Transconductance Amplifier
OVR	Output Voltage Range
PID	Proportional-Integral-Derivative (controller)
RD	Recoverable Degradation
SAT	Satisfiability Problem of Boolean Formulas
SBD	Soft Breakdown
SPICE	Simulation Program with Integrated Circuit Emphasis
SR	Slew Rate
TE	Technology Effects
TDDB	Time Dependent Dielectric Breakdown
TMR	Triple Modular Redundancy
VHDL	Very High Speed Integrated Circuit Hardware Description Language
WCTDT	Worst Case Task Distribution Time

1

Introduction

The continuous spreading of embedded System-on-Chips interacting with its environment is inevitable. With the increasing dependency on embedded electronic devices, the failure susceptibility has to be minimized. However, new requirements like upcoming applications or new hardware platforms have to be faced and handled by the embedded system with equal consistency as the already known failure sources, impeding the efforts to minimize the failure susceptibility. The electronic systems need to be designed highly dependable and robust to execute their assigned tasks reliable. Further, any real-time bound needs to be hold, if the system is operated in a live-critical environment.

Therefore, designing reliable mixed-signal architectures using unreliable hardware is the key challenge. The unreliability is caused by different effects. Just increasing the integration density leads to degrading process reliability and device aging already. The proneness to production failures increases, permanently affecting the system behavior. Also, decreasing the technology size intensifies the failure susceptibility further, as do environmental effects. Any variation of the performances and the system behavior could eventually lead to failing components or failing task executions.

To handle or avoid failures, erroneous behaviors and performances affecting the system, counter mechanisms are necessary. Additionally, for embedded

System-on-Chips those counter measurements need to apply autonomously during run-time, not affecting any real-time bounds. Hence, many reliable systems were proposed mainly for digital systems, not taking mixed-signal processing into account, let alone any kind of analog components.

Therefore, the goal of this thesis is to design a reliable analog architecture providing hardware redundancy and a dependable task distribution system. The cores are held redundant, forming a multi-core design with respect to mixed-signal processing. The task distribution system enhances the system to be dependable and robust to failures. Further, the reliability is increased by minimizing (to the maximum extent of eliminating) the single points of failure.

1.1 Analog Circuit Design Flow

To design analog circuits from scratch is a challenging task. It starts with defining the designs' functional specification and ends at the physical layout, which satisfies the defined specifications. To meet the complexity of this task, several design steps in between are required to partition the design hierarchically into solvable subtasks. The design steps were classified in [GK83], the abstraction levels and domains of the design process were defined with a graph called Y-chart. Until nowadays, the application of the Y-chart has been edited and further defined for the complex analog design tasks [GDWL92, HBKK94]. The three axis of the graph illustrates three views of the design:

Functional Domain: The functional domain describes the temporal and functional circuit behavior in different levels of abstraction. The design starts at the uppermost level, the concept layer, with the definition of the specification of the circuit. Descending to the algorithm layer, functional models are defined to describe the intended analog circuit behavior in a top-level and abstract manner. At the macro layer the transfer functions are determined by the desired behavior models, while the differential equations of the lowermost layer specify each component in detail.

Structural Domain: The structural domain specifies the system and subsystems and the interconnections of all the devices. The behavior describing models at the block layer are mapped to top-level building blocks. Those blocks are either generated from scratch (a top-down design flow) or taken out of existing libraries (a meet-in-the-middle design flow). Descending the abstraction levels to the devices, all interconnections are being defined. Further,

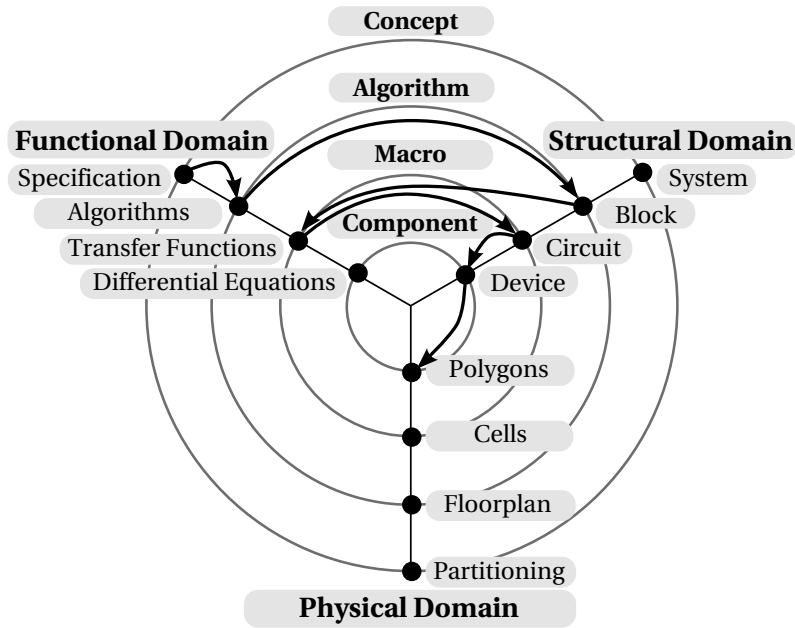


Figure 1.1: Y-chart of the Analog Design with Exemplary Synthesis Steps [Ste11]

the size and the topology of the devices determine the behavior/transfer function of the circuit components. The proper selection of the device parameters classify the circuit as *specification is met*.

Physical Domain: At the physical domain the geometric properties of the system and all their components are defined. At the uppermost layer a general partitioning of the system occurs, followed by placing of the devices. The lowermost layer implies the layout generation, the implementation of the devices and their topology as polygons. With the layouted circuit the physical structure is realized in silicon, ready to go into production.

The different abstraction layers map the design flow from the top-level concept to the low-level components and all the needed steps in between. Descending layers, increasingly defining the details of the design, is called synthesis. Contrary, to verify the conformance of lower levels with higher levels even upon different domains is called analysis. Figure 1.1 shows the Y-chart with the analog design flow, beginning with the functional specification at the top-left side, showing a typical design flow of an analog circuit. For the final layouted design and the manufactured circuit, the specification is used as benchmark to be tested against.

1.2 Circuit Reliability

The analog design still suffers from the major flaw of non-standardized analog design specification, huge design space and missing abstraction like Boolean algebra. So far, standardized components of analog circuit libraries are non existing. Operational amplifiers (OpAmp) or current mirrors (CM) are highly versatile and can be optimized in many directions, such as a high slew rate (SR), no (almost non) offset or many others. This variety hinders standardization and leads to the need to design analog parts almost always from scratch. In Chapter 5.1 follows a detailed list of the specification needed of the analog circuit components used for this thesis, indicating the need to design all components from scratch.

Over the last years, research identified different critical areas of analog circuits, which are classifiable by their failure severity. Also, the rising complexity of analog circuits due to increasing process variation and shrinking technology size cause further sensitive areas. Those areas within the design process call for either a full design verification [GDWM⁺08, GMDW11] to ensure reliability or reliable circuit enhancements to counteract the identified failure mechanisms [BGL⁺06] (Definition 1.2.1).

Definition 1.2.1 (Failure Mechanism)

Failure mechanisms are the physical progress of the failures (which are described by abstract failure models).

Typical circuit enhancements are increasing the transistors size to minimize failure occurrences, or monitoring circuits, which detect failures prematurely and allowing countermeasures to apply. Figure 1.2 structures a set of failure classes with their identified failures [Phe06]. Technology effects and design failures are confronted at the design process for example through verification methods [GMDW11]. Degradation and environmental effects, however, can not be countered by verification methods, but are handled during run-time. They are parted into two distinct categories, Definition 1.2.2 and 1.2.3:

Definition 1.2.2 (Soft Failure Effects)

Soft failing effects are physical effects, which influence/worsen the behavior of the circuit (in regard to the defined specification).

Definition 1.2.3 (Severe Failure Effects)

Severe failure effects are physical effects, which, if occurring, result in the total loss of functionality of the circuit.

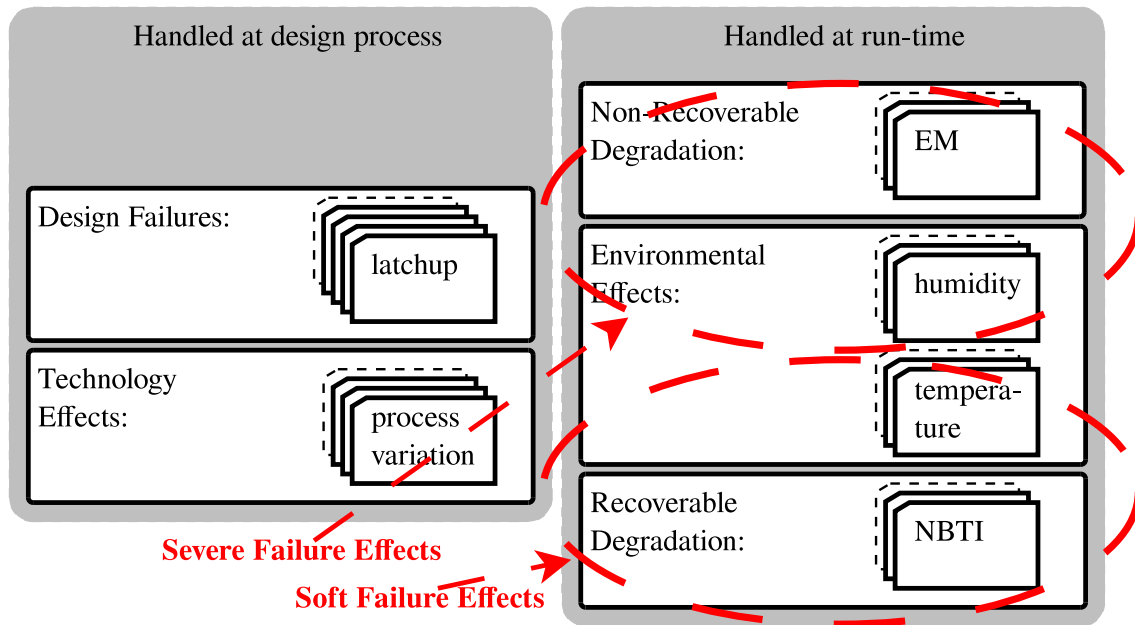


Figure 1.2: Failure Severity Diagram

Soft failure effects are monitorable during runtime and the initial behavior recoverable, if according countermeasures are applied. Since severe failure effects must be prevented under all circumstances, the countermeasures need to be applied during the design process or monitor circuits must be able to detect failures before they occur. The following, detailed description of Figure 1.2 outlines several failure mechanisms:

Design Failures: Until now the design process is mostly done by hand. Faulty and erratic design drafts are common, calling for the need to verify the designs to eliminate any self-made design failure. Typical design failures (DF) are shortages, latchups, leakages, current crowding and cross talks.

- The latchup effect is a shortage caused by a parasitic CMOS structure, which acts as two stacked thyristors (a P-N-P-N structure) keeping themselves in saturation and creating a low-impedance path between the two input signals [RCN04].
- Affected by leakage are capacitors, semiconductors and interconnects, increasing the power consumption leading to the total circuit loss eventually [Phe06, NC10].
 - Charged capacitors are gradually discharged by the attached components, since even in power-down mode some components con-

duct small amounts of current. Also, imperfect or damaged dielectric materials of the capacitor lead to the flow of a leakage current, a constant loss of energy [NC10].

- Semiconductors suffer from the phenomenon of charges tunneling through the insulating regions or the source and drain terminals (called subthreshold conduction). The thickness of the insulating regions determines the leakage current flow, the amount of current, which is lost [NC10].
- Current crowding is an effect evoked by a nonhomogenous current density distribution through the (semi-)conductors, potentially leading to thermal runaways or electromigration, if not addressed properly during the design process [GAY89].
- In analog designs, the effects of crosstalk are defined as the capacitive effects a signal has upon a nearby signal. Crosstalked signals may falsify any output, the circuit is getting useless. The most common prevention methods of crosstalk are increasing the wire spacing and sizes, as well as a reordering of the wires [VMS97, VCMS⁺99].

Design failures are prevented by the design verification through the Design Rule Check (DRC), the Layout Versus Schematic (LVS) test and the parasitic extraction (for the final simulation runs) or other design verification methods [GMDW11].

Technology Effects: Next to the design failures are the failures caused by the chosen technology and the fabrication process, for example the following:

- Process variation,
- Shrinking technology size,
- Shortening production time cycles.

Equivalent to design failures, design verification methods are needed to prevent these technology effects (TE) [GMDW11]. Other methods, like extensive simulation (Monte-Carlo simulations), may eliminate the failures caused by the technology, yes this precludes the desire to shorten the production time cycles.

Non-Recoverable Degradation: Severe impacts on analog circuits are caused for example by *Electromigration* (EM) or *Time Dependent Dielectric Breakdown*

(TDDDB), abruptly ending the lifetime of the circuits [BGL⁺06, PWMC07, CLL⁺07].

- TDDDB „is a measure of how long a dielectric can preserve its high resistivity under thermal and electrical stress.“ [HL12, p. 127] TDDDB occurs as a failure mechanism in MOSFETs, if those are not operating within their specified operating voltages [BGL⁺06, Lie06]. As a result of the long-time application (beyond the operating voltage), the gate oxide is tunneled, forming a conducting path to the substrate [YFB⁺09] and [HL12, p. 127-145], destroying the MOSFET.
- EM is a failure mechanism of the conductors. „Current flow through a conductor produces two forces to which the individual metal ions in the conductor are exposed. [...] The second force [...] is generated by the momentum transfer between conduction electrons and metal ions in the crystal lattice. This force works in the direction of the current flow and is the main cause of electromigration.“ [Lie06, p. 39] The gradual movement of the metal ions in the direction of the current flow are causing the transport of the material. „This depletes the metal of some of its atoms upstream, while causing a buildup of metal downstream.“ [KK11, p. 31] The upstream thinning and the downstream buildup leads to two worst case failure state: an open or a short circuit [Lie06].

It is to state that already progressed degradation by the two mentioned failure mechanisms is irreparable. The degradation progress can be slowed down by, for example, reducing the current-density, but can never be recovered (from which the name derives: non-recoverable degradation (NRD)).

Environmental Effects: Heat and radiation are the most commonly mentioned environmental sources, which influence analog circuits and cause failing effects. However, other environmental effects (EE) also exist. Following, a couple of environmental failure effects are introduced:

- Humidity effects provoke shortages, moisture absorption is either done by the package and the circuit or monitors power-off the affected areas to dry out.
- Hydrogen affects the conductors, inducing metal breakdowns similar to EM.

- High temperature degenerate the lifetime of the circuits drastically (accelerating degradation and EM), interrupted only by cool-down phases. Temperature changes can lead to thermal runaways, ending in a destructive manner.
- Radiation/Ionization effects can be classified into two mechanisms, which affect MOSFETs [SM88]:
 1. The *Total Ionizing Dose* is the cumulative damage worsening the performance over the exposition time. The radiation affects the gate insulation layers of MOSFETs. „Radiation-induced trapped charge has built up in the gate oxide, which causes a shift in the threshold voltage [...] If this shift is large enough, the device cannot be turned off, even at zero volts applied, and the device is said to have failed by going depletion mode.“ [OM03, p. 483] This applies not only for N-type MOSFET, but also for P-type one, where the shifted transistor threshold is never again met.
 2. The *Displacement Damage* characterized the displacement of the atoms of the crystal lattice caused by high energetic particles. [SM88] The resulting change of the electrical property of the devices can cause latchups.

Devices exposed to radiation environments as for nuclear industries or for deep space missions are specifically made radiation hard. Radiation hardness is achieved through specific design, material selection and fabrication methods [SM88]. At the digital domain the equivalence to failures caused by radiation are Single-Event-Effects.

Recoverable Degradation: Recoverable degradation (RD) effects are classified as soft failure effects. Threshold voltage drifts of transistors are recoverable and affect the circuits over time [CB05, DLS09, YFB⁺09, vRSH⁺15]. Those drifts are caused for example by:

- *Hot Carrier Injection* (HCI): „Over time, charge carriers (electrons for negative, or n-channel, MOSFETs; holes for positive, or p-channel, MOSFETs) with a little more energy than the average will stray out of the conductive channel between the source and drain and get trapped in the insulating dielectric. This process [...] eventually builds up electric charge within the dielectric layer, increasing the voltage needed to turn the transistor on. As this threshold voltage increases,

the transistor switches more and more slowly.” [KK11, p.31] and [BGL⁺06, YFB⁺09]

- *(Positive/Negative) Bias Temperature Instability ((P/N)BTI)*: „Whenever you apply voltage to the gate, a phenomenon called bias temperature instability can cause a buildup of charge in the dielectric [...]. After that gate voltage is removed, though, some of this effect spontaneously disappears. This recovery occurs within a few tens of microseconds [...].“ [KK11, p. 31] The quick recovery phase rises the difficulty to observe BTI effects [SGRG10]. Next to the threshold voltage drifts, a decrease of the transconductance and the drain current of the transistor manifests [JRSR05, BGL⁺06, KCS].

Degradation/aging effects can be treated, if detected, through recovery phases or decreasing the stress level of the circuit or its affected components. However, the according failure models and monitor circuits for the degradation detection, especially detecting BTI effects, are subject of current research as [SH11, vRSH⁺15] state.

So far, a common method to tackle the environmental and degradation effects is to overdesign the analog circuits by increasing the transistor length and width greatly. Figure 1.3, based on [QS08, Figure 1], illustrates the change of the circuit parameters over time of different circuit designs. The red colored area is defined as non-acceptable circuit behavior due to the parameter decrease. It shows that overdesign and reliability monitoring with the appropriate recovery methods increase the device lifetime [QS08, HG14]. The paper states clearly, as do [JRSR05, SH11, ALHS12], the need for reliability measurements of mixed-signal systems. For example, [SH11] and [ALHS12] propose failure models to predict the degrading voltage drifts over time. To counter such degradation Figure 1.4 shows that a recovery phases of the stress signal reduces the probability for defect. The proposed failure model in [SH11] uses a stress signal, for example an arbitrary sine, as input. The output calculates the degradation parameter as the weighted sum of the probabilities for defect.

Further, in [JRSR05, YFB⁺09] failure mechanisms for NBTI, TDDB and HCI are presented. [Phe06] focuses on the impacts the scaling of the technology size has upon the circuitry and different failure mechanisms, clearly clarifying the challenge to increase reliability on the device level. However, this thesis focuses not as much upon the different failure mechanisms, but more on the different monitor circuits to detect the different failures. A survey of monitor circuits is

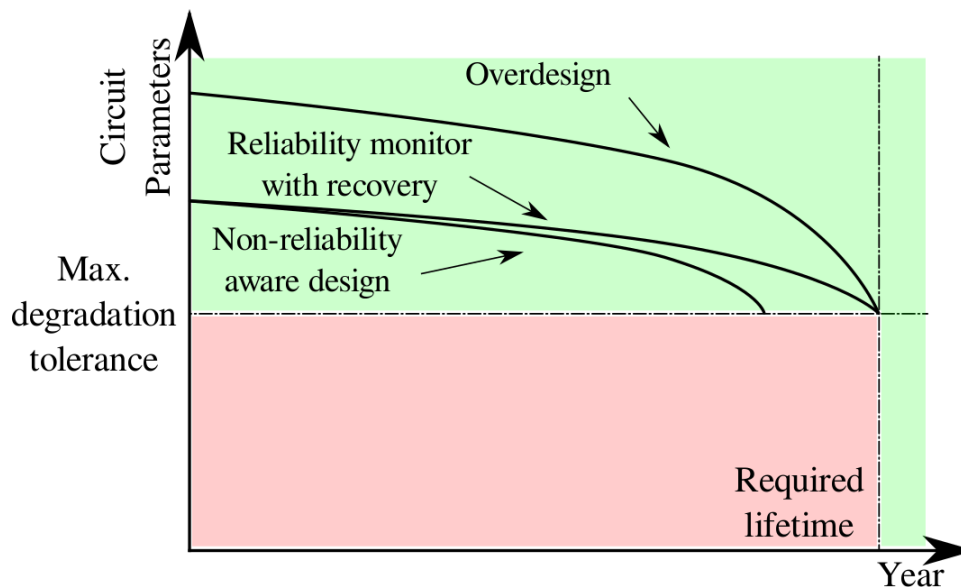


Figure 1.3: *Circuit Performance Degradation of Different Reliability-Aware Approaches based on [QS08, Figure 1]*

provided in [GALH08] with the distinction in offline and online monitoring. Further, the diagnostic values of the different monitor methods are given, as well as the strengths and weaknesses, allowing a quick, rudimentary comparison of self developed monitor circuits with already existing monitor circuits.

Definition 1.2.4 (Online Monitoring)

Online monitoring is defined as guarding the behavior of circuits, subparts and/or single transistors on-the-fly, while the circuit is operating.

Offline monitoring differs only within the definition that the circuit is turned off and the monitoring process has unlimited timing resources. Online monitoring (Definition 1.2.4) is preferred in all applications, which are constantly running, since the device does not have to be powered down or taken out of service. E.g. monitoring the health condition or keeping up predictive maintenance of a circuit is done while running. Incipient failures are detected and predefined actions are taken to minimize or even prevent any downtime. However, it has to be mentioned that online monitors are complex circuits itself, added to every existing circuit they monitor [GALH08]. Further, [SBCD04] states that some failure behaviors are more difficult even impossible to detect online, compared to offline tests. Typical fields of application of online monitors [GALH08, p. 4131] are:

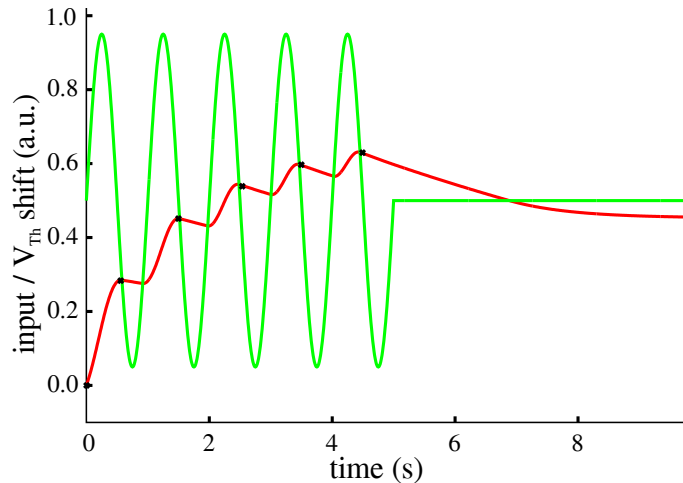


Figure 1.4: *Sine (green) Input Signal with Resulting Probability for Defect V_{Th} shift (red) [SH11, vRSH⁺15]*

- Temperature Monitoring,
- Condition Monitors and Tagging Compounds,
- Current monitors (e. g. power management, charging).

In [QS08] several difficulties of circuit monitoring are stated, whereas the important issue is of how monitor circuits actually experience the same stress, the same failing effect as the monitored devices. Each device, each component, each transistor experiences varying levels of stress, indicating monitor circuits for each of those, almost impossible to realize (because of e. g. process variation), even if the huge monitor overhead is set aside. Therefore, monitor circuits are assigned to groups of neighboring components, keeping the monitor overhead at an appropriate level. Compared to overdesigning, as shown in Figure 1.3, where the change in size of each transistor affects the overall performance of the circuit, calling for precise simulation results of the whole system, monitoring needs only to be done at the most critical areas of the circuits [QS08].

Also, an implementation to monitor NBTI is presented in [QS08], which mitigates the degradation effect by forward biasing the PMOS transistors. The approach introduces a failure mechanism for NBTI and provides a monitor to detect such an effect. Figure 1.5 shows an adaptive body biasing design to monitor NBTI. V_{DD} equals the supply voltage, while a second power supply $V_{DDH} > V_{DD}$ for the body effect of P1 is needed, since $V_{out} \stackrel{!}{=} V_{DD}$ applies, if the circuit is not

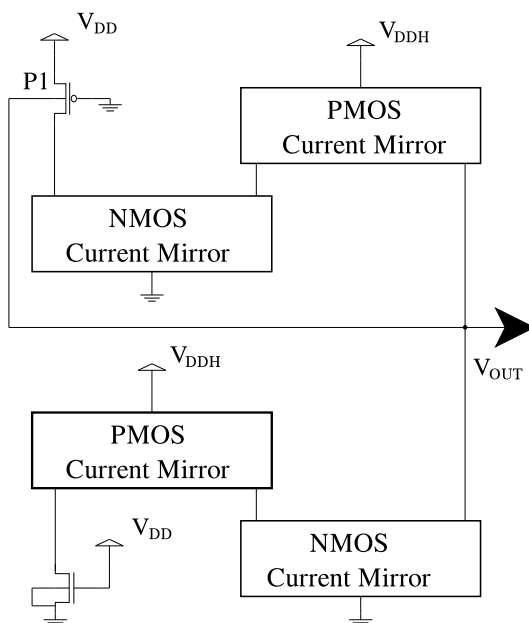


Figure 1.5: *Analog, Adaptive Body Biasing Based NBTI Monitor [QS08]*

stressed. With V_{DDH} being in full swing ($V_{Bulk} > V_{Source}$ at P1) and the gate of P1 connected to ground, P1 is exposed to constant NBTI stress. This influences V_{OUT} , which equals the value of the NBTI monitoring metric, resulting in a decreasing V_{out} with increasing degradation of P1. If V_{OUT} equals V_{DD} no NBTI stress is applied. The approach presented in [SJL08] imposes an on-chip NBTI monitor circuit [SJL08, Fig. 1]. The output of two ring oscillators, one used as reference and stressed only during measurement periods and the other constantly stressed as the monitored circuit, are counted determining the oscillator frequency. The two frequencies are compared, allowing to state the degradation of the constantly stressed ring oscillator.

A circuit implementation to counteract TDDB is presented in [NC13]. The compensation approach is to double the circuit and if the monitor detects any severe breakdown a switching logic turns off the ,old' circuit, while the ,new' circuit is turned on. Yet, the circuit monitoring the soft breakdowns (SBDs), which occur before the catastrophic device failure happen, detectable by unexpected circuit performance variation, is more interesting [NC13]. Fig. 4(a) and (b) in [NC13] show the stress sensor and the reference circuit to monitor sudden performance variations (soft breakdowns). Both outputs are compared to detect the variations.

The on-chip implementation, shown in [KWPK10] catches BTI, HCI and TDDB failures. However, the focus lies on HCI, while TDDB is observed by long-

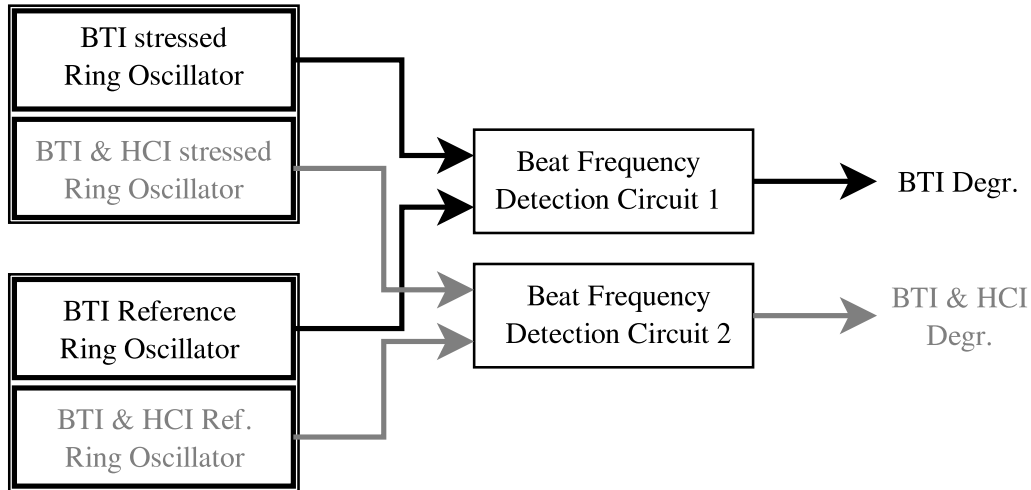


Figure 1.6: *On-Chip Monitor to Detect HCI and NBTI Degradation [KWPK10]*

term stress experiments. The failure detection of HCI is extracted by comparing the monitor results of two different runs:

1. The circuit is stressed so that the degradation is due only to BTI stress.
2. The circuit is stressed so that the degradation is suffered from BTI and HCI stress.

Figure 1.6 illustrates the block diagram to monitor HCI and BTI, using four ring oscillators (ROSC), while two are stressed and the other two unstressed as reference. The actual monitor circuit is the beat frequency detection monitor, seen in [KWPK10, Fig. 11(a)]. Using components like edge detectors and counters, the on-chip beat frequency detection is monitoring the frequency degradation of the ring oscillators.

Lastly, in [EKD⁺03] a monitor is presented to dynamically scale the supply voltage for power-aware computing. The new approach of dynamic voltage scaling is called Razor and is used for embedded, digital processors and SoCs. The dynamic scaling is based on the detection and correction of the timing errors of the circuit. A modified pipeline circuit to recover the delay path failures is also presented. The implementation of the digital monitor circuit is shown in [EKD⁺03, Figure 6], adjusting the supply voltage according to the monitored error rate during operation.

To the best of my knowledge, these are the most important monitor circuits. However, almost all monitor circuits are designed for digital circuits using ring oscillators.

Recalling Figure 1.3 and page 10 the graph shows the increase of the lifetime of circuits, if the critical components are designed reliable (overdesigned). In [HG14] the authors analyzed the lifetime of MOSFETs exposed to BTI stress and how sizing effects and enhances the lifetime. The received perceptions were validated by a yield comparison of an OpAmp to its fabrication. Next, [JRSR05, Phe06] and [YFB⁺09, MDJG12] present analyzes of analog circuits done by specific reliability simulations. Those analyzes led to an increase of the reliability of circuits by detecting and defining the different impacts of failure mechanisms to improve the reliable design of the circuits. [YFB⁺09] proposes a methodology to design reliable circuits and testing them with according reliability simulations. The methodology is partitioned into four steps:

1. The process starts with a new circuit design initially simulated with BSIM models, which are extracted from the technology information.
2. Next to the degradation models in regard to the technology information, the stress voltages are abstracted from the circuit simulation runs.
3. With the degradation models, the circuit simulations and the abstracted stress voltages the circuit failure analysis is carried out, identifying the reliability-critical devices.
4. The analysis results state the lifetime and degradation behavior of the circuits. Further, based on the reliability-critical devices, design improvements are suggested to propose reliable designs.

Another approach is the hierarchical system reliability simulation flow, presented in [MDJG12]. The approach analyzes the reliability of mixed-signal circuits in a hierarchically manner. The flow to improve the reliability of the circuit is as follows:

1. Similar to divide and conquer, the system is partitioned.
2. Every part/subblock is remodeled as a stochastic degradation model indicating the performance evaluation over time of the subblock.
 - The evaluation is done using HCI, BTI and SBD as failure models.
 - To each subblock a sample corresponds, which is characterized by the deterministic input and degradation parameters.
 - The behavior is modeled by a regression model.

3. The complete system level reliability analysis is evaluated using the generated models from each subblock, abstracting the system performances of the circuit.

The gain of reliable designed circuits is measured by comparing the increase of the design time and work load of the two approaches [YFB⁺09, MDJG12] compared to using monitor circuits. An increase of the circuit complexity increases the work load to design reliable significantly. However, reliable designed subcircuits, which are repeatedly appearing within an overall system, would noticeably decrease the monitor overhead to the price of a single time increased work load, a trade-off, which has to be carefully balanced out.

1.3 Reliability-Aware Architectures

Considering the difficulty to enhance reliability and dependability of circuits and systems, a set of criteria classify the different stages of difficulty. The criteria are derived from the conditions the system operates in, the addressed requirements the system needs to fulfill and desired degree of reliability and dependability. The conditions in which the circuit has to operate are for example:

- In a watery or extremely hot environment,
- In an isolated and hard to reach area,
- Exposed to high radiation.

Following are requirements, which the circuit has to fulfill. Examples are:

- Satisfying hard real-time bounds,
- Has to be a low power implementation,
- Minimized in size.

The degree of reliability and dependability for the reliable architecture is defined for example by the following:

- Quality of Service,
- Safety of the system stability,
- Security of the design,
- Needs to be absolutely dependable and reliable (fail-safe), because of operating life-saving devices.

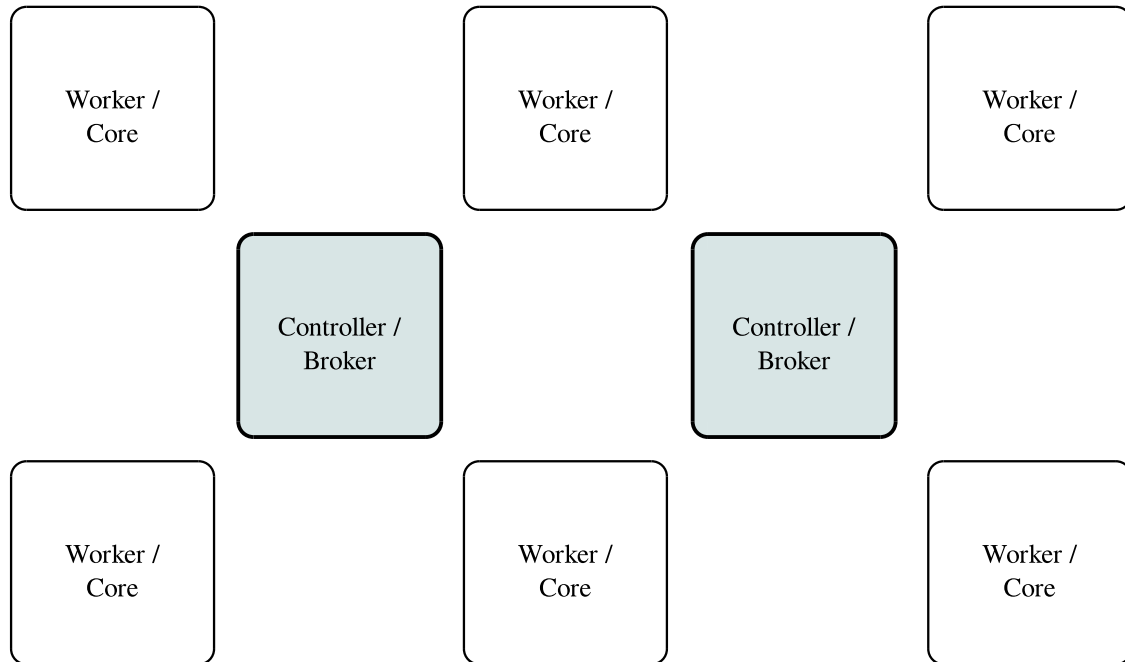


Figure 1.7: *Centralized Reliability-Aware Architectures [BBP13]*

Next to clarify is how the system with its sensors and actuators are interacting with another. The communication can be done by point-to-point connections, by a network to allow advanced broadcasting or basic announcements or simply by signaling paths. Further, those connections are either single ended or advanced to buses to be more robust with a network protocol authorizing re-routing in case of transmission errors. This indicates the possibility of a path monitoring implemented to detect connectivity errors, missing data or the like. All this leads to the specification of a reliable architecture to satisfy the set of criteria defined above. Most commonly implemented are two approaches in the digital domain, which are introduced rudimentary, now. Figure 1.7 illustrates the centralized reliability-aware architecture of the two approaches.

The Multiple Task Distribution Controllers (MTDC) is a redundant, centralized mechanism, regarded to be asymmetric, since cores can acquire two distinct roles: as task receivers (core) or task distributors (controller). Regularly, the controllers send life signs through the system to detect failures within the controllers and determine a leader¹. The cores send health signals (suitability, health state, workload) to the controllers. The elected leader distributes the tasks upon the

¹A leader is in charge of the distribution. While a leader is determined, the other controllers are almost idle, only monitoring working cores and keeping their health signals active

received information. Further, it evaluates neighborhood relations for potential task clustering and watches, detects and counteracts failures. The leader is the only active controller, while the other controllers just maintain the received information to keep coherency with the leader, the global task distribution map.

The Auction-based Multi-Agent System (AMAS) is a partly decentral and redundant approach. It has two roles to distinguish, the role of the broker and the role of the clients and therefore is considered to be asymmetric. The task distribution is done through an auction initiated by the broker. The clients bid according to their suitability, etc. The highest bidder wins. In case of neighborhood relations the broker can decide to distribute the task to another client. Global task information is held by the broker. Self-healing is not intended by default. Life signs have to be send by the clients to the broker, who can detect and counteract failures, additionally. A dying broker is handled by redundant brokers and a frequent broker auction of the leading broker. To be regarded as completely decentralized, every core must be able to be a broker. This rises complexity to be comparable to a centralized task distribution controller, since additional tasks, like initiating auctions, collecting bids, distributing tasks, monitor clients, must be executable at each core.

However, those two approaches are only used as benchmark in Chapter 2 for the three following approaches, which are capable of being mapped into the analog domain.

1.3.1 Artificial Neural Networks

„Artificial neural networks are an attempt at modeling the information processing capabilities of nervous systems.“ [Roj96, p. 3]. Nervous systems as part of neural networks is a branch of neuroscience² [KSJ00, Fin01]. The research has investigated the neural network within brains and based on these discoveries developed computational models, the artificial neural networks (ANNs). The first

²Neuroscience as a field of study dates back to early periods of human history. Evidences are stated that surgical practice on brains have already been performed during the Neolithic times to relieve cranial pressure or curing headaches. However, not until the mid of the nineteenth century extensive neuroscientific knowledge was gained by systematic research, with a significant scientific increase through non-invasive studies of the brain of healthy test subjects [KSJ00, Fin01]. Neuroscience as topic includes a broad range of further studies: Molecular, cellular, developmental, structural, functional, evolutionary, computational, and medical aspects of the nervous system. The techniques have expanded from the individual nerve cells and their composition to complex activities of the brain.

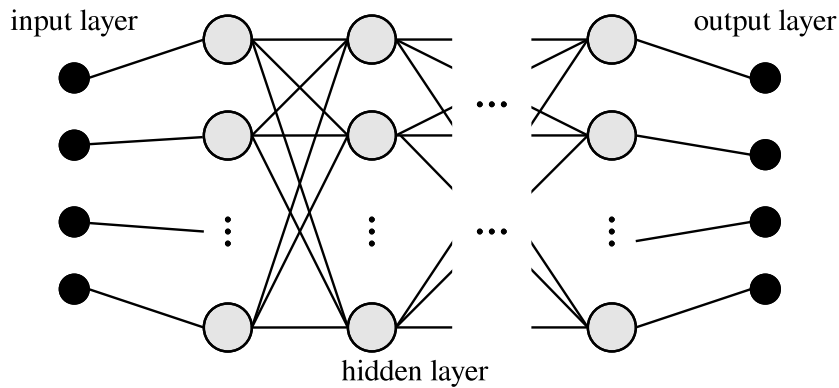


Figure 1.8: *Reliable Architecture using an ANN [Roj96, p.126]*

of such computational models were introduced in 1943 by Warren McCulloch and Walter Pitts and laid the foundation to apply neural networks as instances of artificial intelligence [MP43]. After a period of depression in the 1970s and early 1980s, the neural network research experienced a renaissance in the mid 1980s through associative memory, perceptrons, support vector machines and more recently through deep learning [ZDL90, Hay98].

ANNs are systems of interconnected *artificial neurons*, as illustrated in Figure 1.8. „The input is processed and relayed from one layer to the other, until the final result has been computed.“ [Roj96, p.126] Sensor data for example serve as parameters of the input layer. The intermediate layers with its nodes are called hidden layers, since they are not directly interacting with the external environment [BH00]. „The determination of the appropriate number of hidden layers and number of hidden nodes (NHN) in each layer is one of the most critical tasks in the ANN design.“ [BH00, p. 22] The output layer represents the *network functions* or the tasks the network has to process, all the needed steps for a successful execution are represented by the hidden layer. Adding/deleting the connections between neurons increase the quality of the output [vdM90]. Further, changing the weight of the interconnections results in different network functions/different outcomes of the tasks [Roj96]. The proper selection of the weights, the activation function and the net topology enables ANNs to learn to solve complex non-linear functions and execute various tasks like an autonomously flying aircraft [Cen03]. Therefore, modeling ANNs requires the definition of three important elements [Roj96]:

1. The structure of each artificial neuron (nodes),
2. The topology of the interconnections (network),

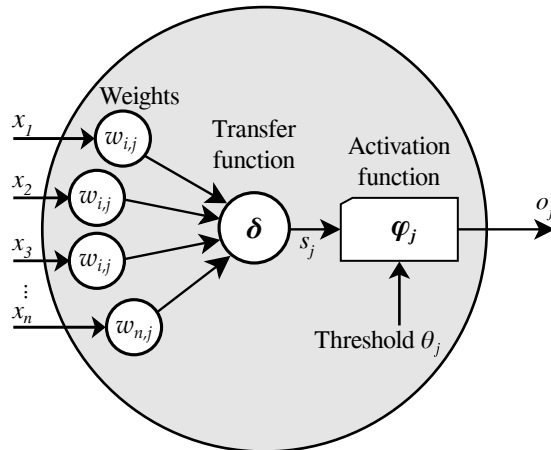


Figure 1.9: Artificial Neuron (based on [Smi97, p.461] and [BH00])

3. The learning algorithm, which weights all interconnections.

Figure 1.9 shows the structure of an artificial neuron (node) [Smi97, BH00], described by its four basic elements:

Weight function: All input parameters $x_i : i \in \{1..n\}$ are weighted $w_{i,j} = [-1..1]$ against each other, defining the ratio of influence each input has upon the neuron. A weight of zero for an input is equivalent to a not-existing edge, neglecting that input and parts the inhibitory influence (negative sign) from the excitatory influence (positive sign) of that input.

Transfer function: The transfer function δ evaluates the overall influence of all inputs, the net value s_j of the neuron. Equation (1.1) states that the input parameter and the weights define the transfer function, but other characteristics of the architecture may also be included.

Activation function: The activation function φ_j evaluates the net value s_j with the threshold value θ_j and determines the output o_j of the neuron. Further, φ_j is defined by the topology of the network and represents the influence each neuron has upon the overall system [MS10]. The functions are usually monotonically increasing, for example as a ramp, piecewise linear or sigmoid, hyperbolic tangent function [SMN11]. Especially in multilayer-perceptron neural networks the sigmoid function is used mostly [Hay98].

Threshold: The threshold value θ_j characterizes the minimum net value of a neuron to be activated, which corresponds to the threshold potential of biological neurons.

The evaluation of the output o_j of the artificial neuron is done by the threshold value subtraction, as equation (1.2) states [Roj96, BH00].

$$s_j = \delta_{i=1}^n (x_i w_{i,j}) \quad (1.1)$$

$$o_j = \varphi_j(s_j - \theta_j) \quad (1.2)$$

In [Mea89, Roj96, MS10] Hardware Neural Networks (HNNs) were introduced, which begin to supersede the numerous software based implementation [MS10]. In [MS10] the development of hardware implementations of ANNs of the past 25 years has been summarized and the advantages of HNNs compared to ANNs stated as:

- The increase in speed by taking advantage of hardware parallelism,
- The decreasing of costs by lowering for example the component counts and power requirements and
- The ability the counteract degradation through fault tolerance and keeping the system running with reduced performances.

Digital architectures are using shift registers, latches or memories to store the dynamic changeable weights $w_{i,j}$ and threshold value θ_j , and look-up tables, standardized adders and multipliers are used for the neuron architecture [Roj96, MS10]. The advantage lies within the simplicity and the scalability (the cascability and flexibility) of the components, the high signal-to-noise ratio and cheap fabrication [MS10]. The analog architectures are using resistors, charge-coupled devices, capacitors and floating gate transistors to store the dynamic changeable values [Roj96, MS10].

Learning involves updating the weights $w_{i,j}$ dynamically, while the size of the components are fixed, which is done by varying the stored charges. Further, those architectures benefit from the physical effects of currents and voltages and are in general optimized in size [Roj96]. However, „obtaining consistently precise analog circuits, especially to compensate for variations in temperature and control voltages, requires sophisticated design and fabrication“ [MS10]. Hybrid architectures represent mixed-signal implementations of the ANN as shown in [SLM99, SMN11]. Their focus is to combine the advantages of both domains, while the weaknesses are minimized.

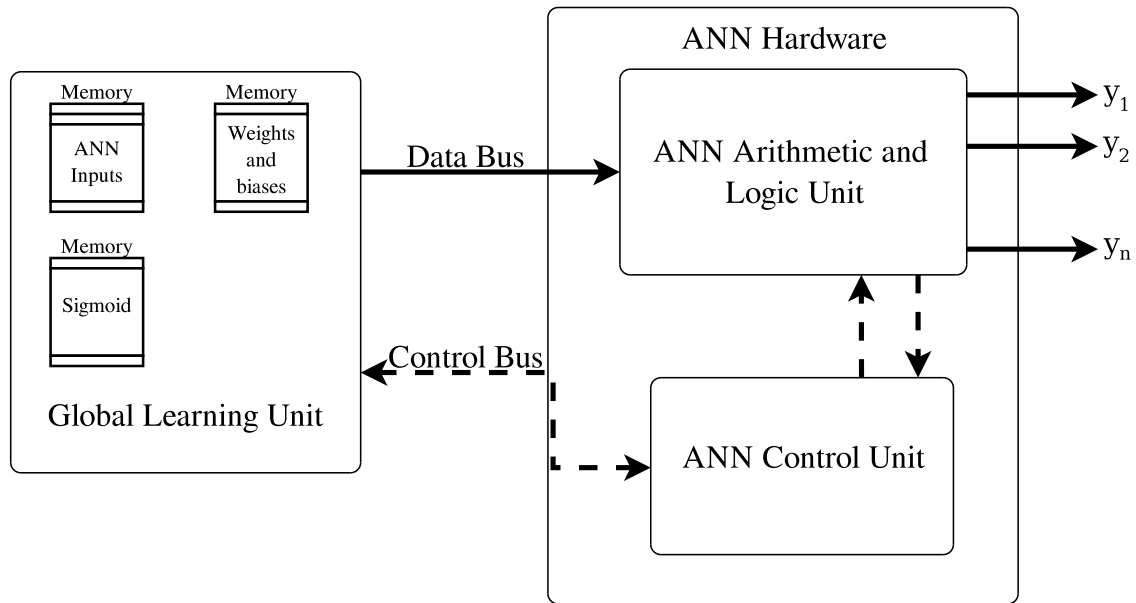


Figure 1.10: *Implementation of an Artificial Neural Network [SMN11]*

In [SMN11] a detailed description of a hardware implementation of the ANN based on Field-Programmable Gate Arrays (FPGA) is provided. Next to the presented hardware layer with the neuron architecture, a global ANN learning unit is needed to transfer the learning effects provided onto the hardware. „The control unit [...] commands the digital neuron arithmetics only.“ [SMN11, p. 655] The split between the control and the arithmetic unit allows the hardware to adapt to different multilayer perceptron neural networks [Roj96, SMN11] topologies, since the number of inputs and the dynamic changeable values can be altered on-the-fly. Therefore, the hardware is able to perform different ANN applications. Figure 1.10 shows the block diagram of the ANN hardware shown in [SMN11].

The strength of ANNs lies within machine learning and classification/ranking, achieved by the learning ability of neurons. The weights and threshold values can be chosen randomly at the beginning. Even a basic trial-and-error learning algorithm adjusts the values until the wanted behavior/result occurs. Therefore, a period of learning effectively affects the output of an ANN [Roj96]. „The problem is the time required for the learning process, which can increase exponentially with the size of the network.“ [Roj96, p. 451] ANNs were successfully embedded in computer vision and speech & pattern recognition [Mea89]. A further field of application of ANNs is within robotics, with focus on reliability and specialized actuators/manipulators like prosthesis [MS10].

The strength of learning is also its weakness. ANNs tend to memorize the training data and blind themselves by that, unable to re-adjust to new data [BH00]. Only carefully chosen net topologies prevent this overfitting [BH00], often referred to as the bias-variance trade-off [GJP95, EPP00, SS01]. Also, the training data need to be collected or generated manually [Pom93, Roj96]. It is essential, that a large training diversity of real-world operations is maintained [Pom93]. For this thesis the learning unit is regarded as circuit black box and not further investigated. Furthermore, to compare ANN to the other approaches, ANN has to be abstracted to be used as a task distribution system.

ANN as Task Distribution System The most basic approach is to add the neural network as a centralized distribution unit containing for example $m \cdot N$ -many neurons. The neural network is connected to the global learning unit and to each working core, as illustrated in Figure 1.11(a). Each core has its own monitor supplying the life signs to the neural network, while the output signals of the neural network unit represent the *task_{i on}* signals of each core.

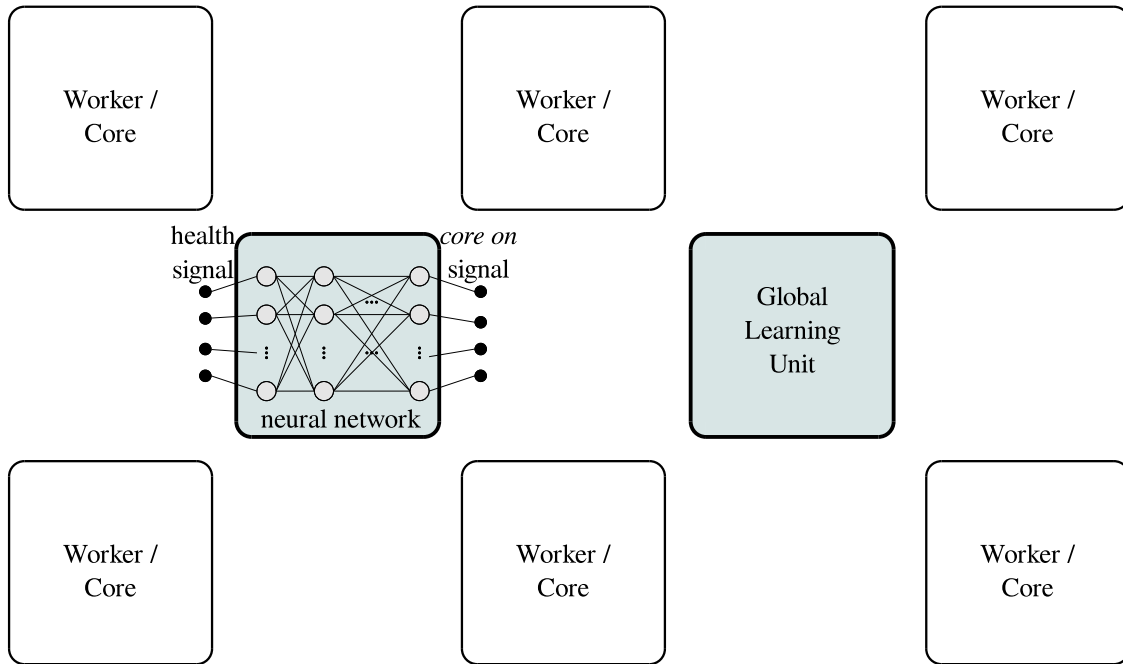
The other approach, seen in Figure 1.11(b), is slicing the neural network into pieces and equips each core with an appropriate slice. This leads to the following:

1. The neural network is dispersed over the system and therefore less likely to be destroyed by a single impact, increasing the reliability.
2. The slices may differ in size, each core only needs those neurons placed locally, which are capable to trigger the core to allocate an appropriate task.

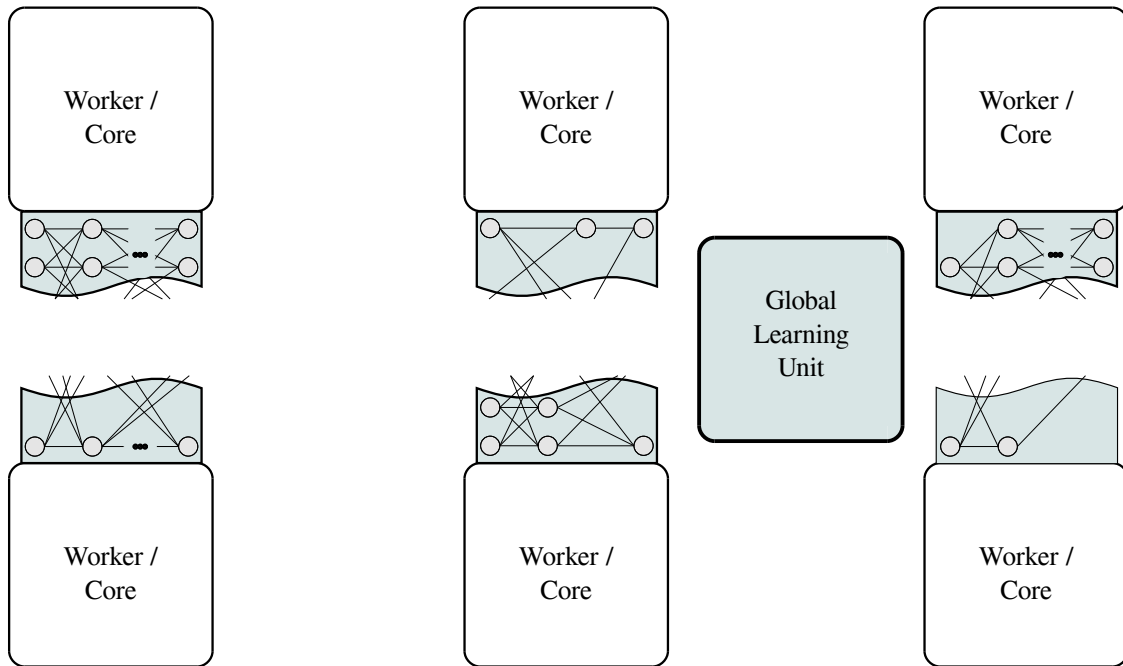
The global learning unit is neglected in regard to symmetry of the mechanism. The unit is used beforehand offline to determine the weights $w_{i,j}$ and threshold value θ_j , but remains idle during the online operation time. The semi-decentralized symmetric is used for any further consideration regarding ANN.

1.3.2 Analog Voting

Redundancy of circuit components increases reliability also, since the loss of one redundant component has no impact on the overall functionality of the analog circuit. Though, the workload or task of the lost component has to be reallocated and carried out by other components, demanding a control unit or architecture to distribute the tasks or balance out the loads over the different components. The approach presented in [NAN08, ADSN09, AN11] aims at the concept of redundant components to obtain reliability. An analog voter (AV), similar to a broker



(a) *Centralized Asymmetric Approach*



(b) *Semi Decentralized Symmetric Approach*

Figure 1.11: *Artificial Neural Networks as Task Distribution System*

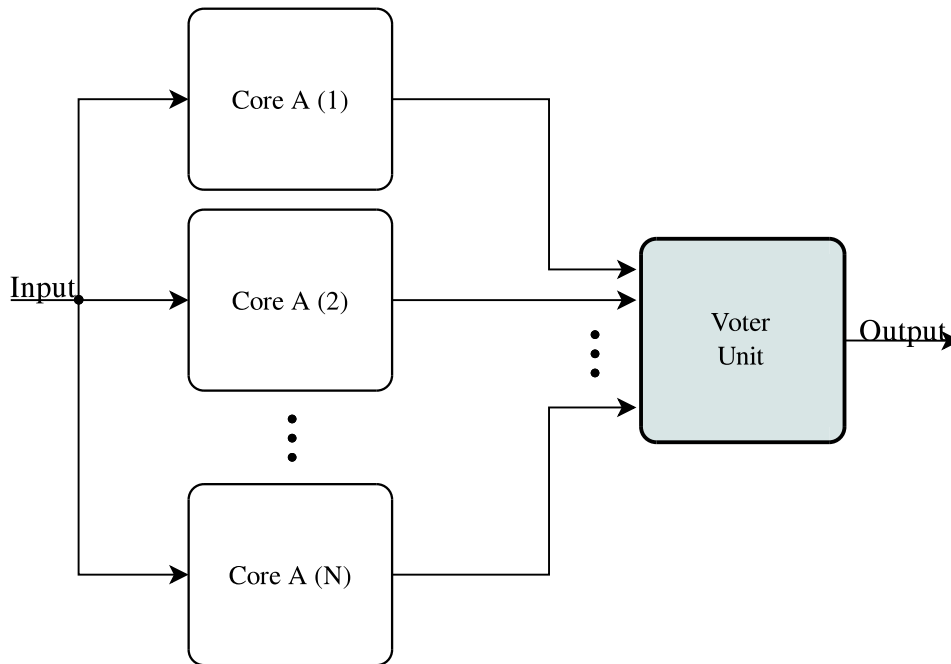


Figure 1.12: *Reliable Architecture using AV [AN11]*

or controller, is used as decision unit. Each core and $N - 1$ replica of it suitable for the task are executing the task and send their output to the voter. The voter then decides for each output signal upon all received data, which is the best result and passes it on. The algorithm, deciding on the best result, is either a majority, a mean or a median vote for N -tuple modular redundancy (NMR) systems [NAN08, AN11]. Figure 1.12 shows an NMR system used as a reliable architecture. The key question for an analog NMR voting system is, which parameters to be voting for. The approach published in [NAN08] uses the „amplitude as the voting parameter. [...] the amplitude comparison itself can be used to detect problematic metrics such as gain, propagation delay and slew rate altogether.“ [NAN08, p. 335] However, other specifications, like bandwidth, are not caught by this voting parameter [NAN08].

1. The majority voter compares all inputs and chooses the value, which is equivalent by more than half of all inputs. For a triple modular redundancy (TMR) system the majority vote follows equation (1.3) [NAN08].

$$V_{out}(t) = \begin{cases} In2(t), & \text{if } |In1(t) - In2(t)| \leq \Delta V, \\ In3(t), & \text{otherwise.} \end{cases} \quad (1.3)$$

Using the majority voting algorithm, core failures or any other unexpected overshoots or significant outlier within the output of some cores are simply ignored [NAN08].

2. The median value is not as easily calculated, but is picked from the collection of inputs. Sorting the inputs through a set of comparators and subtractors identifies the middle value, which equals the median. The order of the comparators and subtractors for any number of inputs is defined by the sorter tree, published in [Par91]. Equivalent to the majority voting algorithm, core failures or any other unexpected overshoots or significant outlier within the output of some cores are simply ignored [NAN08].
3. The mean voter is a modification of the algorithm published in [ADSN09] and implemented as an N -input transconductance cell voting on voltage values only. In [AN11] the schematic is published and described in detail and proves that the output voltage V_{out} is calculated by equation (1.4), the mean of the function:

$$V_{out} = \frac{V_{in1} + V_{in2} + V_{in3} + \dots + V_{inN}}{N} \quad (1.4)$$

The mean voting algorithm is not as robust against single overshoots or significant outlier, but process variation, aging effects and the like are counteracted by the mean evaluation [AN11].

For $N = 3$ the majority vote is the best choice in regard to the amount of comparators. However, for any $N > 3$ the count of needed comparators $C_{AV}(N)$ increases significant compared to the median voter, as Table 1.1 shows [NAN08]. For the mean voter, as published in [AN11], the amount of needed transistors

Table 1.1: Majority, Median and Mean Voter Costs [NAN08, AN11]

N	Majority Voter $T_{AV}(N) \approx 20 C_{AV}(N)$ [NAN08]	Median Voter $T_{AV}(N) \approx 20 C_{AV}(N)$ [NAN08]	Mean Voter $T_{AV}(N) = 2 + 4 N$ [AN11]
3	20	60	14
5	80	160	22
7	600	280	30
9	1700	440	38

$T_{AV}(N)$ need to be compared to $C_{AV}(N)$. A standard comparator uses approximately 20 transistors [Bak10].

The advantage of the analog voting approach is the very fast, inexpensive and linearly scalability of the $N - 1$ replicas to meet the failing core tolerance [NAN08]. However, as with brokers, the loss of the voting unit represents a total loss of the system. A second voter would invalidate the single point of failure, but needs to be controlled as well, including a third master voter, deciding, if voter A or voter B is taken. „Voter size and its probability of failure can be ignored in a low defect rate technology [...]“ [AN11, p. 2], but in nanoscale technology the voter reliability can not be neglected. The master voter must not fail, but be highly reliable. Voters are not suitable for other tasks. The trade-off is reliability versus costs, costs caused by overhead (size) and additional needed power.

1.3.3 Artificial Hormone System

Similar to neural networks, the artificial hormone system (AHS) is bio-inspired. As a reliable architecture the focus lies on redundant cores connected to sensors and actuators and a task allocation system to distribute the work load [vRBP11a, vR12]. Figure 1.13 in combination with 1.15 show the basic concept of using AHS as a reliable architecture.

To enable a high level of redundancy a generalized core and task concept is issued, as seen in Figure 1.14. A detailed description of the generalized core concept is published in [vRSH⁺15]. The concept allows to simplify even highly specialized architectures with an active processing core and re-active components like memories, timers, amplifiers and converters. The simplification results in a heterogeneous multi-core architecture interconnected through an artificial hormone system. Further, generalizing the functionality of cores and tasks also allows a high level of flexibility and reliability in terms of dynamic adjustments and system stability. Hence, AHS focuses on organizing its sensors and actuators and distributing tasks [vRBP11a]. Each core has a hormone communication module/hormone decision unit and bases its decision of taking a task upon the hormone level of that task. The behaviors and activities feasible by this architecture are referred to as tasks, which are executed on suitable cores.

Figure 1.15 illustrates the hormone balancing loop and the three hormone types [BPvR08, vR12], affecting another. The decision unit is attached to the workers/cores and has i numbers of decision modules applying for $i \in m$ different tasks. A monitor is guarding the decision unit and the cores, control-

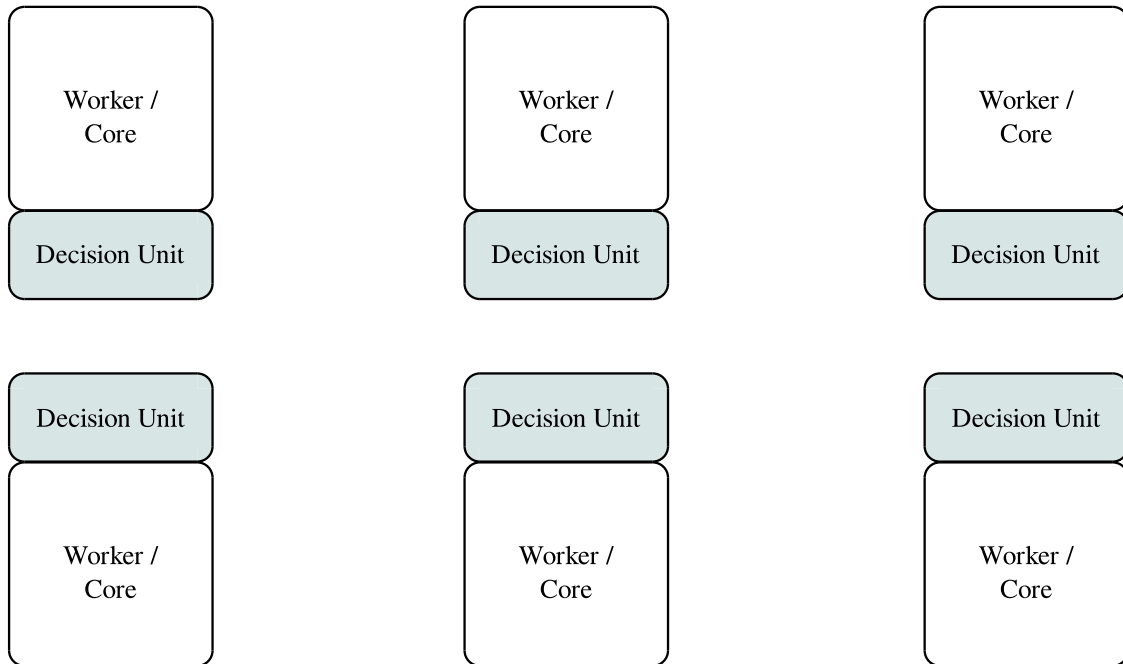


Figure 1.13: *Reliable Architecture using AHS*

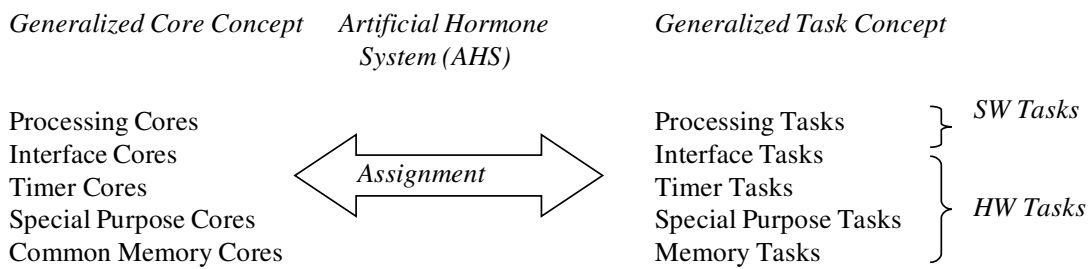


Figure 1.14: *Assignment of Generalized Tasks to Generalized Cores by AHS [vRSH⁺15]*

ling the eagerness of the core to apply for the different tasks. Each core calculates the hormone level of a task for himself, based on the three hormone values [BPvR08, vR12]:

Eager Value: The eager value represents the eagerness of a core to a specific task, simultaneously representing the suitability of the core. The higher the value, the more eager the core and therefore the better the suitability [BPvR08].

Suppressor: The suppressor hormone suppresses the suitability and therefore the eagerness of a core to take a task. It is subtracted from the eager value [BPvR08].

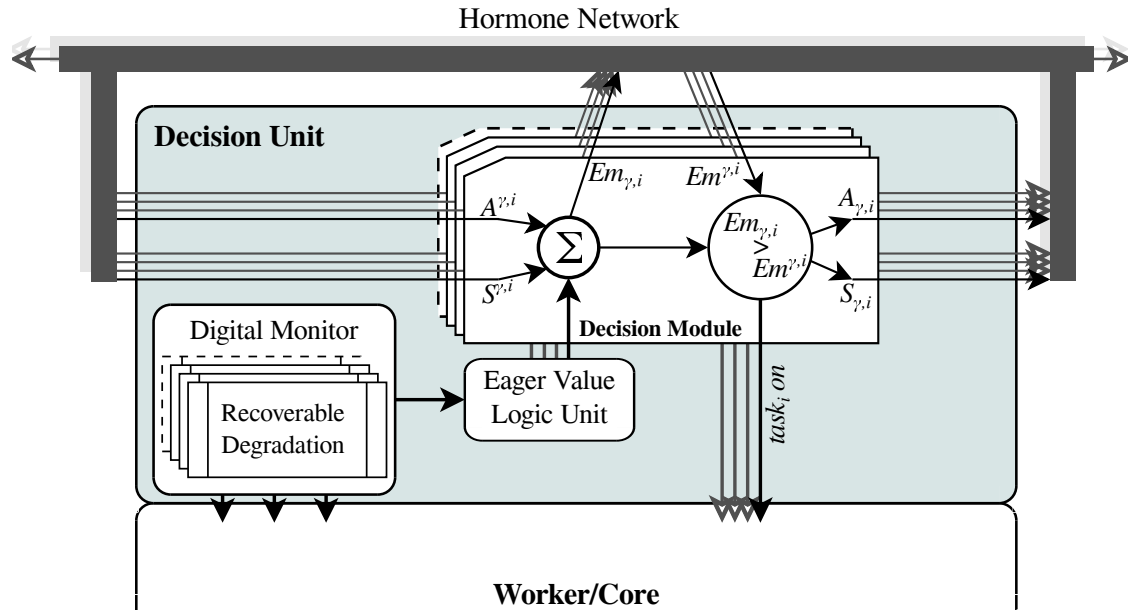


Figure 1.15: Artificial Hormone Loop [vRSH⁺15]

Accelerator: Contrary to the suppressor, the accelerator raises the suitability. This hormone type is added to the eager value, favoring the task allocation [BPvR08].

A balanced hormone loop of a task indicates that the task is allocated. Occurred failures or any other mischief imbalances the loop, stimulating all cores and re-initializing the task allocation process. To ensure the stability of the hormone loop, the sum of suppressors must be greater than the sum of accelerators [BPvR08]:

$$\forall i : \sum_{\gamma=1}^N S_{i\gamma} > \sum_{\gamma=1}^N A_{i\gamma}. \quad (1.5)$$

Therefore, a cyclic evaluation of the hormone loop is needed. [BPvR08, vRBP11b] analyzed the time behavior and the stability condition of the allocation cycle. Further details on the hormone loop are presented in [BPvR08, vRBP11a, vRSH⁺15].

The failure handling is either a re-active behavior, the standard method, a failure occurs and the system reacts, or a pro-active behavior, a guarding monitor triggers a safety routine and a task switch is initiated [vRSH⁺15]. A task loss caused by a failure comes hand in hand with the total loss of data or progress of the task. A restart of the task at a new core is the only option. However, the

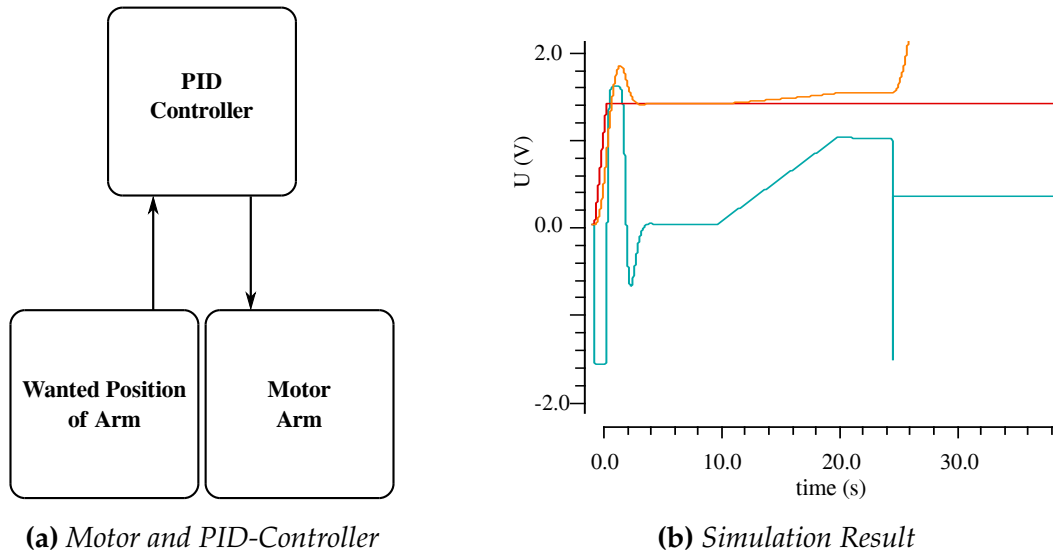


Figure 1.16: Failing of a Motor Control using a PID-Controller, displays the *wanted positions* (red), the *actual positions* (orange) and the *PID controlling* (cyan)

pro-active behavior allows to save the task state and any data attached to it (common memory cores), reallocates the task to a new core and continues at the saved state. Therefore, the failure handling through the balanced hormone levels, as an imbalance indicates a task reallocation, ensures the dependability of the architecture.

1.4 Motivating Example

The following motivating examples show the need for reliable hardware to assure a fault-tolerant and dependable execution of tasks, especially if those tasks are life-saving and/or are framed by hard real-time bounds. The first example is an actuator, a Proportional-Integral-Derivative (PID) controller, to control a motor steering a gripper arm. Figure 1.16 shows the simple hardware implementation and the resulting simulation with the failing PID controller.

The second example is a sensor, which is low-pass, band-pass and high-pass filtering acoustic signals, set for two different cut-off frequencies. Figure 1.17 shows the implementation and the simulation results of a slowly increasing offset at one amplifier, which leads to a failing low-pass filtering. Any output at the speaker dies, as the filter dies.

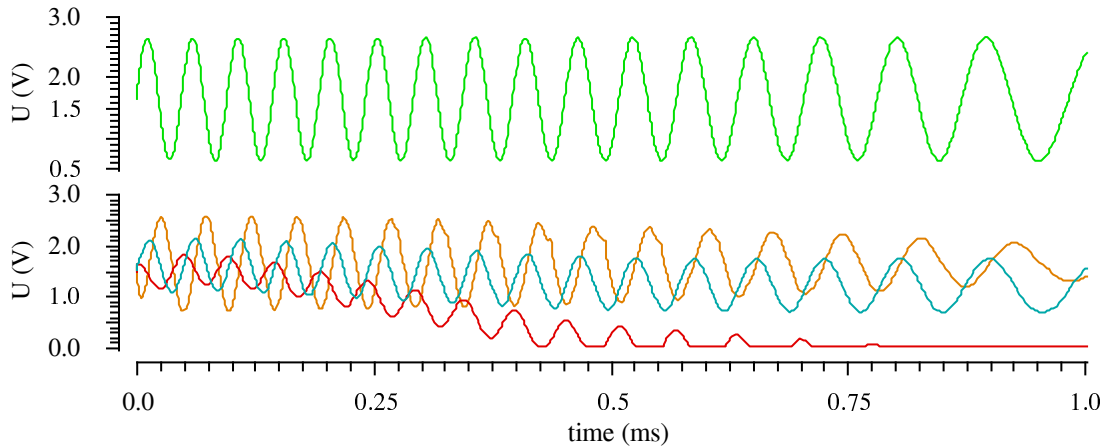


Figure 1.17: *Failing of a Signal Filtering*

- *Strip 1* displays the input *sine (green)* signal
- *Strip 2* displays the three output signals *high-passed (orange)*, *band-passed (cyan)* and *low-passed (red)*

Within the scope of this thesis an analog architecture is developed to ensure the distribution and the execution of tasks on a reliable architecture. A failure analysis, introduced in Chapter 4.2, states the feasibility and the reliability of such an approach, all depending upon the monitor circuits, introduced in Chapter 5.4, identifying failing cores. The failing task execution seen in Figure 1.16(b) and 1.17 is caught by the architecture (the monitor circuits) and a redundant core reallocates the task and continues its execution.

It becomes clear that those two motivating examples can be expanded to maneuver a drone, for example a quadcopter, making sure the rotor control stays active and filtering the video input to follow a track. Without the PID controller any flying maneuver is not feasible, the quadcopter loses control and crashes. With the analog distribution hardware another PID controller allocates the rotor control and keeps the drone flying (though a quick regulation with little turbulences might be the case). Even further, a robot in deep-space can keep itself active and running to fulfill the mission it was send on.

1.5 Publications

Parts of this thesis have been published in [vRBBH12, vRH12, BvRHB13, vRSH⁺15, vRMH15], while [LPB⁺12, SvRH14] were done in collaboration and

had an impact on this thesis. The late publication of [BvRHB13] introduced the idea of an analog and reliable architecture, while a first analysis and estimation of timing and area constraints were shown at the VLSI-SoC PhD forum [vRH12]. In [vRBBH12] the approach is manifested by a first implementation of a prototypic schematic and presented simulation results. The major impact on the approach has [vRSH⁺15], evolving and deepening the voltage and current-based architectures and presenting failure models and the according monitoring circuits. The publication of [SvRH14] presents a first approach of a failure analysis and of a symbolic modeling of the implementation. A symbolic analysis leading to the specification of the architecture, which was then fully designed and fabricated, has been published in [vRMH15]. Also in [vRMH15] the first measurements of the silicon prototype were presented.

1.6 Overview

The remaining chapters of this thesis are organized as follows. A comparison of five reliable architectures is done in Chapter 2, followed by Chapter 3, which introduces the new reliable, analog architecture developed in this thesis. The preliminaries, which are significant to fully design the analog circuitry of such an architecture, are characterized in Chapter 4. Further, a failure analysis classifies the architecture in regard to reliability and dependability. The synthesis of the design with the resulting schematics, their layouts and failure monitoring is presented in Chapter 5. Chapter 6 discusses the results obtained by operating the architecture and classifies timing constraints and other specifications. Finally, the thesis concludes with Chapter 7 and provides in addition suggestions for future work.

2

Comparison of Reliable Architectures

As already mentioned at the beginning of Chapter 1.3 the five approaches presented are classified and compared in several categories, which are essential for this thesis to achieve the envisioned objectives.

Each approach is suitable to act as a task distribution system, the Multiple Task Distribution Controllers (MTDC) and Auction-based Multi-Agent System (AMAS) use a centralized unit to distribute the tasks, ANN uses its learning unit to acquire the neuron network structure to distribute tasks, the AV has each task executed multiple times and decides upon the best result and AHS enables a decentralized distribution, all which are defined as the distribution mechanisms.

The self-configuration property is stated, if a system is able to distribute the assigned tasks to cores for execution. self-optimization is obtained, if states like health, workload of neighborhood relations of all cores, including newly participating cores, are considered during the distribution process. Self-healing indicates the handling of task drops or core outages or similar events, which lead to task reallocation. Self-reliance implies active monitoring, able to counteract failures by provided failure mechanisms, which influence the health states and the like to fully detach the system from human influence. If those four self-x properties apply, the system is in a reliable state of self-control.

The needed size overhead in terms of dependability and the gain in reliability are state for each approach. However, they are closely related to each other, since for example maximizing F_t indicates an inevitable increase of size overhead O . The goals are

1. minimizing the needed size overhead O ,
2. maximizing the failing core tolerance F_t to state the reliability gain.

Lastly, the ability to migrate task, which can be executed at both domains, between analog and digital cores is stated. Summarizing, the categories, in which the approaches are compared, are:

- Task distribution mechanism,
- self-control obtained by
 - self-configuration,
 - self-optimization,
 - self-healing,
 - self-reliance.
- Size overhead in terms of dependability,
- Reliability gain,
- Real-time bounds (assigning m tasks),
- Mixed-signal task migration.

A first comparison of AHS, MTDC and AMAS has already been published in [BBP13]. Figure 2.1 sketches the differences between the five proposed reliable architectures.

- The AHS architecture is shown in Figure 2.1(d). The distribution mechanism is integrated into each core, each equal in their contribution to the system. As long as active cores are capable of performing the required tasks the system is operating and tasks are distributed. Therefore, this architecture is considered to be symmetric.
- In Figure 2.1(a) the centralized reliability-aware architectures (MTDC and AMAS) with a redundant set of task controller is illustrated. The distribution is done only as long as one task distribution controller is active, but fails if all controllers are breaking down, since controllers only distribute

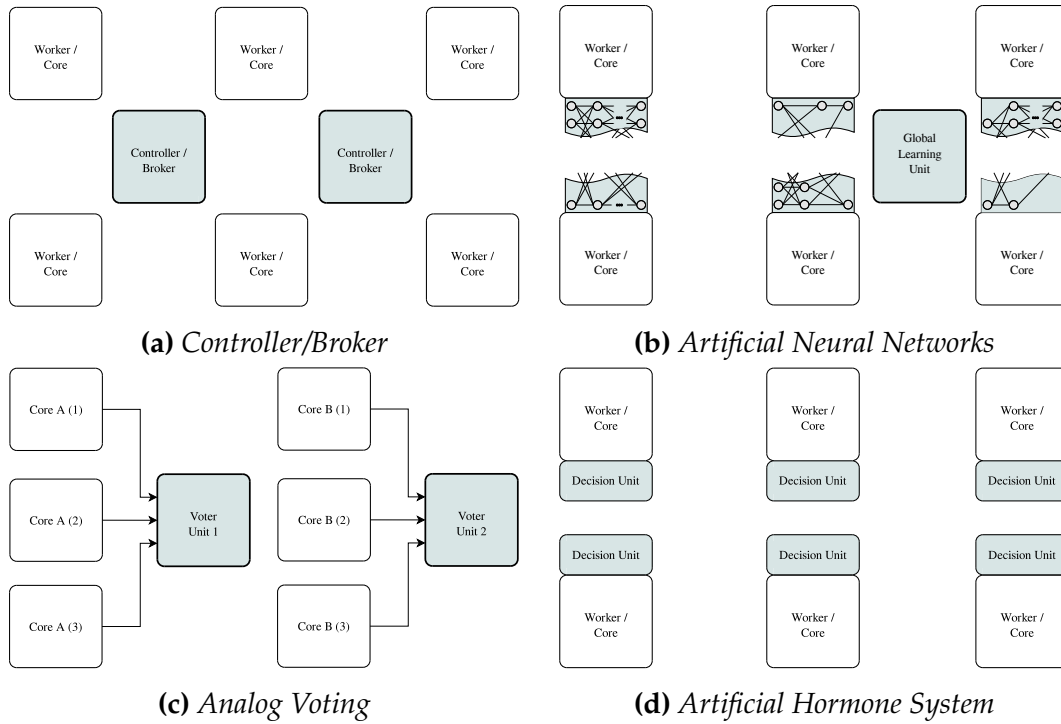


Figure 2.1: *Different Reliability-Aware Architectures*

tasks, while the remaining cores just execute them. Due to the two distinct roles, this distribution architecture is regarded as asymmetric.

- Figure 2.1(c) shows the AV architecture, which needs to be regarded as a centralized reliability-aware architecture. The results of the task execution is only passed on as long as a master voting unit is active.
- The ANN architecture is considered to be a reliability-aware architecture, since some neurons and single cores can fail without affecting the overall system output, and as decentralized, if the neural network unit is sliced appropriately and distributed as decision units onto each core, as seen in Figure 2.1(b). However, the fully connected neural network structure is dispersed over the chip and is highly dependent on the result of the learning and its training data.

2.1 Size Overhead

Comparing the sizes of the different reliable architectures is rather difficult, since usually digital architectures are measured by the amount of required gates and analog architectures by the amount of required transistors. Further, it has to be stated that analog cores may differ significantly in size, though the specifications vary only little. Table 2.1 shows the increase in size, which applies to the overall system, caused by the reliability-aware architecture. To simplify the stated size increase, the size of monitor circuits are not taken into account. The global routing is neglected as well due to its small impact on the overall size, even though - or because - the needed routing of all five approaches differ significantly. The centralized controller and the auction-based distribution system are summarized and presented by the subscript CB. Also, for AHS and ANN the averaged percentage increase of the size at every core is specified.

Table 2.1: Size Increase of the Different Reliable Architectures

- _{AHS}: Size of the distribution mechanism, which is added to every core
 ⇒ Decision modules of the AHS requires 31 232 gates [vRSH⁺15]
 ⇒ Percentage increase of each core $\square_{\%AHS} = \frac{\sum_{i=1}^N (\square_{\text{core}_i}^{\text{AHS}})}{N}$
- _{CB}: Size of the controller or broker
 ⇒ Controller instances [CM08] require 278 345 NAND-gates [BBP13]
- _{AV}: Size of the voter to vote for N cores
 ⇒ Size of the needed comparators and subtractors (see Table 1.1)
- _{ANN}: Size of the learning unit and the neuron architecture, which is added to every core, see Figure 2.1(b)
 ⇒ Percentage increase of each core $\square_{\%ANN} = \frac{\square_{\text{learning unit}} + \sum_{i=1}^N (\square_{\text{core}_i}^{\text{ANN}})}{N}$

To neglect the difficulty caused by the sizing differences, an overhead comparison in terms of reliability is done, also ignoring the monitor circuits and the global routing. The overhead comparison is oriented on the two centralized approaches MTDC & AMAS and defines the value at which the challenger is still to favor. The higher the value, the more overhead still surpasses the size increase through a centralized controller and favors the challenger. For the following comparison a couple of variables have to be defined:

- N defines the total amount of cores, while N_f defines the maximum amount of failing cores compensated by redundancy [BBP13].
- N_{CB} represents the amount of controllers of a centralized task distribution system by which the total amount of cores N increases (see Figure 2.1) [BBP13]. N_W represents the amount of working cores.
- Similar to N_{CB} , the amount of voters of the AV are represented by N_{AV} . The minimum amount of needed voters is displayed at equation (2.1)¹ increasing the total amount of cores N ,

$$N_{AV,\min} = \begin{cases} 1 & \text{if } 3 \leq N < 9, \\ \sum_{k=1}^{\lfloor \log_3 N \rfloor} \lfloor \frac{N}{3^k} \rfloor & \text{if } N \geq 9, \end{cases} \quad (2.1)$$

wherein $3 \leq N$ is the least number of cores for voting to apply (Table 1.1).

- N_{ANN} represents the amount of neurons, while $N_{ANN} > N$ since $N_{ANN} \approx m \cdot N$ applies with m equals the amount of tasks.
- In any cases $\{N, N_f, N_W, N_{CB}, N_{AV}, N_{ANN}\} \in \mathbb{N}^*$ and $N_f < N$ has to be true.

One objective is to minimize the needed size overhead O , which is highly depending on the failing core tolerance F_t . Equation (2.2) states the percentage calculation of allowable failing cores to the total amount of cores, which defines the failing core tolerance. As special case $F_t = 0$ (equation (2.3)) needs to be defined, stating that a single core outage will result under worst case failing condition into a full system failure, leading to Definition 2.1.1.

Definition 2.1.1 (Failing Core Tolerance)

The failing core tolerance determines under worst case condition the probability to withstand core outages, while maintaining the full system functionality, which is given by equation (2.2).

¹The equation is based on Table 1.1 and the least number of cores to be voting for. With for example nine cores each three cores are connected to their own voter, while those resulting three voters are checked by one final decisive voter, totaling the number of voters to four. The amount varies for every number of cores divisible by three, and so on. Further details are given in Appendix A.2.

$$F_t = P(X = \text{,WC failing core(s)'}) = \frac{N_f}{N} : 0 < F_t < 1. \quad (2.2)$$

$$F_t = 0 \Rightarrow \text{special case} \quad (2.3)$$

Following, the operation constraints are defined to state the reliability even under worst case failing conditions. If the according running requirement are fulfilled the architectures are defined as *operating reliable*:

$$\text{running}_{\text{CB}} : N = N_{\text{CB}} + N_{\text{W}} \mid N > N_{\text{W}} + N_f, N > N_{\text{CB}} + N_f \quad (2.4)$$

$$\bigwedge N_{\text{CB}} \geq N_f + 1, N_{\text{W}} \geq N_f + 1 \quad [\text{BBP13}] \quad (2.5)$$

$$\Rightarrow N \geq 2N_f + 2 \quad (2.6)$$

$$\text{running}_{\text{ANN}} : N > N_f \mid N = N_{\text{W}} + 1, N_{\text{W}} \geq N_f + 1 \quad (2.7)$$

$$\bigwedge C_{\text{learning unit}} \in N \quad (2.8)$$

$$\text{running}_{\text{AV}} : N = N_{\text{AV}} + N_{\text{W}} \mid N > N_{\text{W}} + N_f, N_{\text{W}} > N_{\text{AV}} \quad (2.9)$$

$$\bigwedge N_{\text{AV}} \geq N_f + 1 \quad (2.10)$$

$$\text{running}_{\text{AHS}} : N > N_f \mid N = N_{\text{W}}, N_{\text{W}} \geq N_f + 1 \quad [\text{BBP13}] \quad (2.11)$$

The overhead for the centralized task distribution systems is calculated by the amount of controllers with respect to the total amount of cores [BBP13] and is shown in equation (2.4). Further, equation (2.5) allows to define the lower bound for the overhead calculation of the centralized approaches, stated in equation (2.14).

$$O_{\text{CB}} = \frac{N_{\text{CB}}}{N} \quad (2.12)$$

$$O_{\text{CB}} = \frac{N_f + 1}{N} = F_t + \frac{1}{N} \quad [\text{BBP13}] \quad (2.13)$$

$$\text{Lower bound: } O_{\text{CB},\text{min}} = \frac{N_f + 1}{2N_f + 2} = \frac{1}{2} \quad (2.14)$$

$$\text{Upper bound: } O_{\text{CB},\text{max}} = \frac{N - 2}{N} \quad (2.15)$$

The upper bound of the needed overhead is stated in equation (2.15) and is derived from equation (2.5), too. Furthermore, equation (2.6) clearly state the need for $N \geq 4$ for any centralized architecture, otherwise the core failure tolerance is stated as $F_t = 0$. Also, it is to assume that $N_{\text{CB}} \leq N_{\text{W}}$ applies, anything else is not applicable.

Showing that $O_{CB} \prec O_{AV}$ applies: Similar to O_{CB} , the overhead of AV is calculated by $O_{AV} = \frac{N_{AV}}{N}$. For $N < 9$ the overhead is reduced to one master voter $O_{AV} = \frac{1}{N}$, which equals exactly the overhead of the centralized approaches, but following $N \geq 9$ of equation (2.1) the size overhead of AV increases faster, stating that

$$O_{CB} \preccurlyeq O_{AV} \quad (2.16)$$

applies. A comparison of \square_{AV} and \square_{CB} also favors the MTDC and AMAS, since the voting unit increases significantly in size as Table 1.1 on page 25 shows. Only for low N the analog voting approach is comparable in terms of overhead.

Shown that AV loses to the MTDC and AMAS in size and size overhead, the focus lies on the performance of ANN and AHS in regard to the centralized approaches. The overhead of AHS and ANN, in regard to the percentage increase due to the size of the task distribution mechanism on each core, are calculated by

$$\begin{aligned} O_{AHS} &= \square_{\%AHS}, \\ O_{ANN} &= \square_{\%ANN}. \end{aligned}$$

However, O_{AHS} and O_{ANN} differ, because $\square_{\%ANN}$ also depends on the global learning unit, which supervises the learning and dynamically changes the weights $w_{i,j}$ and the threshold θ_j . It can be assumed that $\square_{\%AHS} \leq \square_{\%ANN}$ and therefore $O_{AHS} \prec O_{ANN}$ applies, indicating that the overhead of AHS will be less than the overhead of ANN. The assumption can be validated by the following calculations leading to equation (2.22). The lower bound of the overhead of the centralized controller or broker is defined by $O_{CB,\min} = \frac{1}{2}$.

Showing that $O_{AHS} \prec O_{CB}$ applies: As long as the overhead of AHS is less than $O_{CB,\min}$, the size increase of AHS will always be smaller than the size of a centralized controlling core, as equation (2.17) states.

$$\begin{aligned} O_{AHS} &< O_{CB,\min} \\ O_{AHS} &< \frac{N_f + 1}{2N_f + 2} \\ \square_{\%AHS} &< \frac{1}{2} \end{aligned} \quad (2.17)$$

This applies to any number of failing cores N_f , as seen in Figure 2.2(a). If the overhead O_{AHS} is within the green area, $O_{AHS} \prec O_{CB}$ applies, otherwise no prediction can be made.

Showing that $O_{ANN} \prec O_{CB}$ applies: Equivalent to $O_{AHS} \prec O_{CB}$, if $O_{ANN} < O_{CB,min}$ applies, the size increase of ANN will always be smaller than the size of a centralized controlling core. Equation (2.17) is adjusted to match for ANN, assuming that the learning unit is about the size of a controller or broker (see equation (2.18)).

$$\square_{\%ANN} = \frac{\square_{\text{learning unit}} + \sum_{i=1}^N \left(\frac{\square_{ANN}}{\square_{\text{core}_i}} \right)}{N} \approx \frac{1}{N} + \frac{\sum_{i=1}^N \left(\frac{\square_{ANN}}{\square_{\text{core}_i}} \right)}{N} \quad (2.18)$$

$$\begin{aligned} O_{ANN} &< O_{CB,min} \\ \square_{\%ANN} &< \frac{N_f + 1}{2N_f + 2} \\ \frac{\sum_{i=1}^N \left(\frac{\square_{ANN}}{\square_{\text{core}_i}} \right)}{N} &< \frac{N_f + 1}{2N_f + 2} - \frac{1}{N} \\ \frac{\sum_{i=1}^N \left(\frac{\square_{ANN}}{\square_{\text{core}_i}} \right)}{N} &< \frac{N_f + 1}{2N_f + 2} - \frac{1}{2N_f + 2} \\ \frac{\sum_{i=1}^N \left(\frac{\square_{ANN}}{\square_{\text{core}_i}} \right)}{N} &< \frac{N_f}{2N_f + 2} \end{aligned} \quad (2.19)$$

The fraction $\frac{\sum_{i=1}^N \left(\frac{\square_{ANN}}{\square_{\text{core}_i}} \right)}{N}$ represents the size of the neural network averaged over the number of cores N . Figure 2.2(b) shows the blue region in which $O_{ANN} \prec O_{CB}$ applies.

Showing that $O_{AHS} \prec O_{ANN}$ applies: To state if $O_{AHS} \prec O_{ANN}$ the relative complements of O_{AHS} and O_{ANN} need to be viewed. The two relative complements are defined by equations (2.20) and (2.21) and shown in Figure 2.2(c).

$$\{O_{ANN}\} / \{O_{AHS}\} = \emptyset \quad (2.20)$$

$$\{O_{AHS}\} / \{O_{ANN}\} \neq \emptyset \quad (2.21)$$

Hence,

$$O_{AHS} \prec O_{ANN} \quad (2.22)$$

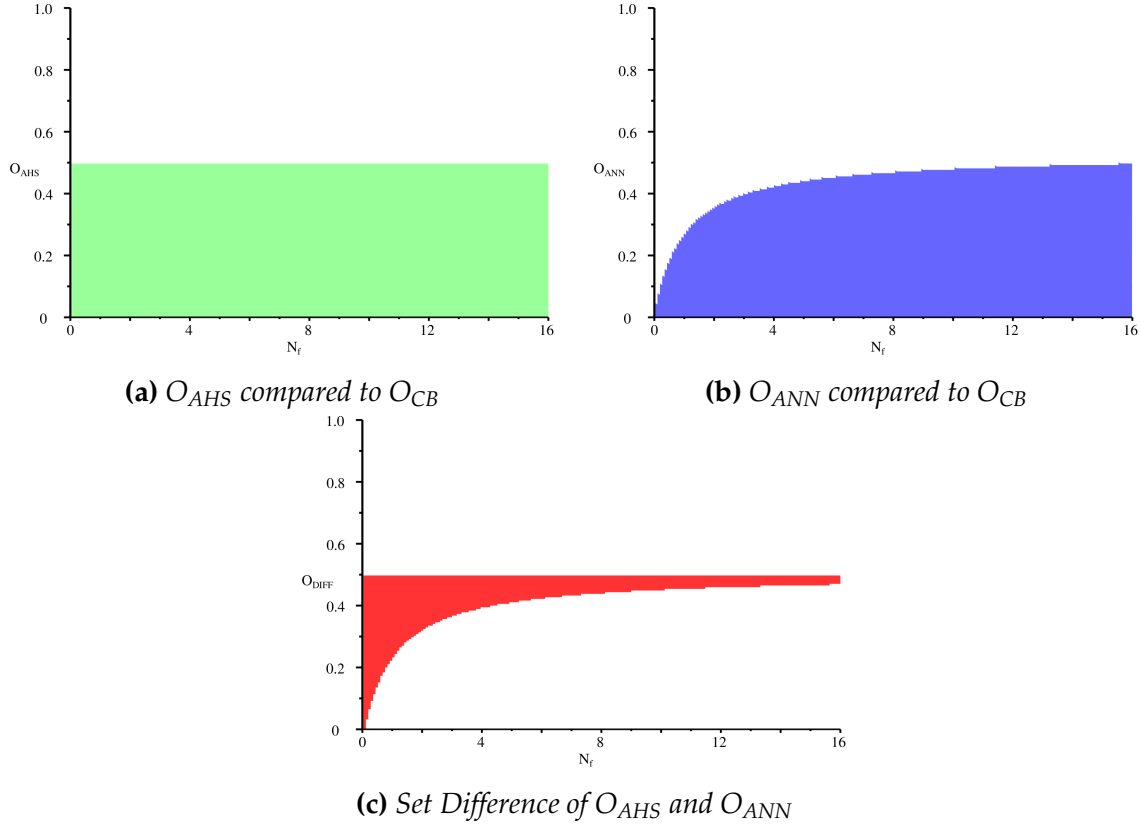


Figure 2.2: Allowed Overhead of ANN and AHS Compared to the Centralized Brokers

is true. The difference of allowed overhead of $\frac{1}{2}$ to $\frac{N_f}{2N_f+2}$ to beat the overhead of the centralized controller or broker results from the size of the needed global learning unit of ANN. This applies also to state that $\square_{\%AHS} \leq \square_{\%ANN}$ is true. With low numbers of N_f the learning unit of ANN is not to compensate, while increasing N_f the overhead calculations assimilate in regard to O_{CB} .

According to equations (2.16) and (2.22), a ranking of the four approaches is given in equation (2.23) and places O_{AHS} first and O_{AV} last.

$$O_{AHS} \prec O_{ANN} \prec O_{CB} \preceq O_{AV} \quad (2.23)$$

The ranking reflects the overhead performance of all four architectures. The overhead in size of AV is already equal or worse than the size overhead increase due to the needed centralized controller, and therefore not challenging AHS and ANN in overhead. Contrary, O_{AHS} and O_{ANN} challenge AV and MTDC & AMAS with equal settings due to the case $N < 9$ leading to $O_{CB} \equiv O_{AV}$.

2.2 Reliability Gain

Increasing the reliability of a system can be achieved through several modifications. The most obvious one is the implementation of redundant components, but those must be controlled, rising the complexity and introducing further sources of failures. Next, the controlling units may also be improved to increase the reliability and to eliminate the new failure sources. Similarly increasing the robustness against process variation or aging effects of each component for example, affects the reliability of the system and applies equally to all approaches. In the following, for each of the five approaches their gain of reliability is described:

MTDC & AMAS: The fixed redundancy states clearly the increase of reliability, each task has a fixed number of redundant cores applying to it, only to be differentiated in favor by the advanced role of brokers compared to controllers. Redundant brokers and controllers heighten reliability further, but as centralized units are responsible for monitoring the cores and themselves. Cool-down phases and the like are initiated by the brokers or controllers as a healing period. Failure detection can be done locally at the cores. However, the centralized units are the weakest spots of the distribution mechanism. A high level of reliability of a centralized architecture can only be achieved by significant overhead in communication and implementation to ensure consistency of the redundant components (e. g. to maintain consistent versions of tables for current task/core mapping, task/core suitability, core load and health status, task relationships, etc. which is handled locally and decentralized by AHS).

ANN: The increase in reliability is gained by two steps:

1. The learning unit finds an initial distribution of the task, with the according neurons being activated the first. Well chosen training data allows the prediction, which neuron (and its core) fails first and hence which neuron should be activated to reallocate the task. Expert knowledge can be included through the learning unit, allowing to influence the distribution.
2. The neural network is, once the learning unit has fitted all neurons, reliable and stable. The task distribution is done safely, even though single neurons may fail, leading to core outages and the need to reallocate the task.

The neural network is sliced to represent the different decision units, which are attached to the according cores. Each slice is attached to each core, allowing to raise the failing core tolerance, which corresponds to the core failing factor. The global routing is still neglected for the comparison.

AHS: Contrary to the MTDC and AMAS, the flexible redundancy results in a further increase of the percentage of tolerable failing cores, while the amount of cores is consistent. In [vRBP11b] using AHS has been proven to be highly dependable and reliable. Reliability and robustness of the approach result from almost no single points of failure due to the high level of redundancy. The cores and their decision modules are held redundant, a failure of one does not affect the systems functionality, while overhead in communication and implementation is still low [BPvR08, vRBP11b, vRBP11a]. Additionally, the symmetry with respect to cores and decentralized task distribution indicates that the task distribution mechanism is equally spread over the chip area. This increases reliability, because partially occurring failing effects, like heat or TDDB, can not harm the distribution mechanism itself. As of the self-healing property, failure detection/monitoring and the according countermeasures can be done locally at each core.

AV: The reliability gain of AV is based only on the increase of redundancy and the robustness of the voting units. A failure of a voting unit disables the task, which the unit is assigned for. All cores are permanently active, never cooling down or healing. Further, the increase of voting units to increase reliability still demands a master voter, which theoretically imply a failing core tolerance $F_t = 0$, stated in equation (2.30). The voting units need to be designed as highly reliable circuits.

In [BBP13] the comparison of reliability gain of AHS, MTDC & AMAS is enforced with simulating different scenarios. The scenarios contained 30 or 60 tasks distributed over 20 cores with each core applying to five tasks and either random-based core failures or worst case core failures, like controllers or brokers. For the random-based failures the simulations were repeated 1000 times to obtain reliable results. The results showed that AHS performed best in all cases, followed by the AMAS and lastly the MTDC. Comparing the running conditions of equations (2.4) and (2.11) supports this observation, since the failure tolerance F_t of AHS can be much higher than F_t of the centralized approaches.

Many approaches, just to mentioning a few [BH00, DGLY05, RSS06], exist to improve reliability of structures, predictions and more using ANNs, but none focused on the actual gain of reliability through the ANN as a task distribution system compared to other methods. The increase of reliability depends on the learning heuristic and the adjustments made to the weight functions of the neurons, but as a single point of failure it is a crucial component.

So far, a comparison in size was done with the objective to minimize the overhead in size of all approaches. In the following, F_t is maximized for each approach:

Upper bound of $F_{t,CB}$: First, the failing core tolerance of $F_{t,CB}$ (equation (2.2)) for the centralized approaches is stated at equation (2.24), derived from the equations (2.4) and (2.6).

$$F_{t,CB} = \frac{N_f}{N} = \frac{N_f}{2N_f + 2} \quad (2.24)$$

$$\Rightarrow F_{t,CB,max} = \lim_{N_f \rightarrow \infty} F_{t,CB} = \frac{1}{2} \quad (2.25)$$

Upper bound of $F_{t,AHS}$: The failing core tolerance of $F_{t,AHS}$ for AHS is given at equation (2.26), derived from the equation (2.11).

$$F_{t,AHS} = \frac{N_f}{N} = \frac{N_f}{N_W} = \frac{N_f}{N_f + 1} \quad (2.26)$$

$$\Rightarrow F_{t,AHS,max} = \lim_{N_f \rightarrow \infty} F_{t,AHS} = 1 \quad (2.27)$$

Upper bound of $F_{t,ANN}$: Similar to AHS, the failing core tolerance of $F_{t,ANN}$ for the ANN is given at equation (2.28), derived from the equation (2.7).

$$F_{t,ANN} = \frac{N_f}{N} = \frac{N_f}{N_{W+1}} = \frac{N_f}{N_f + 2} \quad (2.28)$$

$$\Rightarrow F_{t,ANN,max} = \lim_{N_f \rightarrow \infty} F_{t,ANN} = 1 \quad (2.29)$$

Upper bound of $F_{t,AV}$: Lastly, comparing the failure tolerances of AV and the centralized approaches favors the MTDC and AMAS. Limited by one final decisive voter equation (2.10) can never be true, since $1 - 1 \geq N_f$ leads to the special case of equation (2.3): $N_f = 0$.

$$\forall AV : F_{t,AV} = 0 \quad (2.30)$$

$$\Rightarrow F_{t,AV,max} = 0 \quad (2.31)$$

Since $F_{t,AV}$ represents the failing core tolerance under worst case conditions and the voting units are regarded as own cores, it has to be stated that for AV the worst case failing core tolerance equals zero.

Summarizing the different F_t s: Several observations can be seen:

1. $F_{t,AHS,max} \gg F_{t,CB,max}$ applies,
2. $F_{t,ANN,max} \gg F_{t,CB,max}$ applies,
3. $F_{t,AV,max} = 0$ applies,
4. $F_{t,AHS} \geq F_{t,ANN}$ applies, since based on the equations (2.11) and (2.7) the total amount of cores N for AHS and ANN differs.

Therefore, for $N < 4$ the centralized controller or broker and the analog voter have an equal failing core tolerance, which equals zero. Though, the $F_{t,CB}$ of the centralized approaches rises to $\frac{N_f}{2N_f+2}$ eventually.

Summarized, the reliability gain of all approaches is compared in terms of the tolerance of failing cores F_t (equation (2.2)) at Table 2.2. The cells show either the winning (green) or the losing constraint (red). The AHS and ANN are favored in regard to the centralized approaches. The AV places last due to a core failing tolerance of $F_t = 0$. Equation (2.27) implies that AHS is to favor in terms of failing core tolerance. The ANN follows closely behind AHS (equation (2.29)).

Table 2.2: Comparison of the Failing Core Tolerance

chal. \ def.	CB	AV	AHS
AV	$F_{t,CB} > 0$	-	-
AHS	$\frac{N_f}{N_f+1} > \frac{N_f}{2N_f+2}$	$F_{t,AHS} > 0$	-
ANN	$\frac{N_f}{N_f+2} > \frac{N_f}{2N_f+2}$	$F_{t,ANN} > 0$	$\frac{N_f}{N_f+2} \not> \frac{N_f}{N_f+1}$

2.3 Real-Time Bounds

Satisfying the real-time bounds is one of the crucial performances, each task distribution system has to be measured against. Without real-time capability, the system is not suitable for real-world operations. A worst case task distribution time (WCTDT) estimation of m tasks for the five approaches is given below.

The approach presented in Chapter 3 is measured against these real-time performances.

MTDC: The worst case communication time of a controller with its cores equals

$$\tau_{\text{MTDC}} := \text{cal} + 2 \max(\text{comm.time}(\gamma, \kappa)) : \gamma, \kappa \in N \text{ cores}, \quad (2.32)$$

which synonymously stands for a single communication cycle and whereas *cal* represents the calculation time of the distribution map. Within one cycle, the controller can distribute all tasks, but misses on the answers of the cores in regard to task clustering, accelerations and the like. Therefore, the worst case task distribution time receiving all answers equals $1 + (m - 1)$ cycles. The first cycle indicates the initial task distribution calculation and the worst case task distribution time defined by equation (2.33) [BBP13].

$$\text{WCTDT}_{\text{MTDC}} = m \cdot \tau_{\text{MTDC}} \quad (2.33)$$

$$\Rightarrow \mathcal{O}(m) \quad (2.34)$$

AMAS: The auction-based approach has a similar worst case communication time, which equals to

$$\tau_{\text{AMAS}} := \text{auc} + 2 \max(\text{comm.time}(\gamma, \kappa)) : \gamma, \kappa \in N \text{ cores}, \quad (2.35)$$

whereas *auc* is defined by the auction-based distribution time. Equivalent to the MTDC, the worst case task distribution time is given by equation (2.36) [BBP13].

$$\text{WCTDT}_{\text{AMAS}} = m \cdot \tau_{\text{AMAS}} \quad (2.36)$$

$$\Rightarrow \mathcal{O}(m) \quad (2.37)$$

AHS: The worst case task distribution time equals $2m + e - 1$ hormone loops, with $m + (m - 1)$ loops for assigning all tasks and distributing the accelerator hormone, whereas $e < m$ implies the expiration of a hormone [vRB07]. If a hormone is not refreshed within e hormone loops, the hormone is considered outdated. A hormone loop corresponds to twice the maximum communication time of any two cores [vRBP11b]:

$$\tau_{\text{AHS}} := 2 \max(\text{comm.time}(\gamma, \kappa)) : \gamma, \kappa \in N \text{ cores}. \quad (2.38)$$

Detailed descriptions of the timing behavior and stability conditions are published in [vRB07, BPvR08, vRBP11b]. In [BP12] an aggressive AHS approach is published, which ensures a worst case task distribution time of $m \cdot \tau_{\text{AHS}}$ hormone cycles.

$$\text{WCTDT}_{\text{AHS}} = 2m \cdot \tau_{\text{AHS}} + e - \tau_{\text{AHS}} \quad (2.39)$$

$$\Rightarrow \mathcal{O}(m) \quad (2.40)$$

AV: As soon as the system operates, the tasks are executed on every suitable core. The real-time bounds are only depending upon the calculation time of the median, mean or the majority vote [Par91, Par92] for a decision, which result is chosen. The worst case decision time is τ_{AV} and is considered to be the decision/communication cycle, determined by the slew rate of the comparators of the voting unit. The voting indicates a healing process, if it is repeated regularly. Similar to AHS, an expiration period e would ensure a refreshing of the voter decision, leading to the worst case task distribution time shown in equation (2.41).

$$\text{WCTDT}_{\text{AV}} = \tau_{\text{AV}} + e \quad (2.41)$$

$$\Rightarrow \mathcal{O}(1) \quad (2.42)$$

ANN: The learning problem of ANNs is considered to be NP-complete [Roj96] as the learning problem maps on k -SAT. Several heuristics as well as weight initialization strategies exist to increase the speed of learning [Roj96, Cau96], but to hold hard real-time bounds learning periods are poorly suited. On the other hand the learning is done before going online and distributing tasks. Once online, the worst case task distribution time depends only on τ_{ANN} , the time of the neural network, leading to equation (2.43).

$$\text{WCTDT}_{\text{ANN}} = \tau_{\text{ANN}} \quad (2.43)$$

$$\Rightarrow \mathcal{O}(1). \quad (2.44)$$

Comparing the asymptotic upper bound of the five approaches, it can be stated that

$$\mathcal{O}(1) \leq \mathcal{O}(m) \quad (2.45)$$

applies. $\mathcal{O}(1) \leq \mathcal{O}(m)$ is true, since $1 \leq m$ applies.

2.4 Summary of the Comparison

Finally, the results of the different comparisons of the task distribution systems are summarized in Table 2.3. The AHS and ANN are decentralized approaches, spreading the task distribution mechanism equally over the chip area, which indicates an increase of reliability. For AMAS to be regarded as a decentralized, but still asymmetric approach, each core has to be able to be a broker. However, this rises the complexity and therefore the size and computational overhead of each core significantly. The MTDC and AV are centralized approaches, the two distinct roles are assigned to each core at the design process and never changed. Failing cores are never replaced, the tasks allocated to the others. Further, the symmetry/asymmetry of the mechanisms determines the distribution of the decision units over the prototypical chip. A small number of controller or broker reflect an asymmetric architecture.

In regard to artificial intelligence within robotics, the four self-X properties are needed to ensure self-control. The AHS maintains those properties the best. AHS configures itself by allocating the tasks to cores through exchanging hormones. It keeps itself optimized, since tasks are reallocated by changing hormone levels caused for example by decreasing health states, heavy workloads or neighborhood relations. The AHS is considered to be self-healing, since failures, which result in task drops, core outages or similar events, lead to task reallocations. AV has the tasks distributed before the initial start and never reallocates any task, but the voting unit passes on the best results, close to self-optimizing. The self-control property addresses the ANN through the learning period. The centralized approaches MTDC and AMAS hold those four properties for the price of increased complexity of the controller and broker, especially in regard of reliance and the monitoring. Monitoring several cores and declaring the individual health signals regarding different failure mechanisms is highly sophisticated, a complex task of its own.

All five approaches satisfy the real-time bounds, which is essential for real-world application. Three of them are depending on their communication cycle ($2 \max(\text{comm.time}(\gamma, \kappa)) : \gamma, \kappa \in N \text{ cores}$), while AV always runs all tasks simultaneously and ANNs neural network is already configured and reacts as trained. The WCTDT for the AHS is the slowest and for AV the fastest. Only the ANN holds difficulties with the learning period besides $\mathcal{O}(1)$, but existing heuristics even of hardware implementations [SdBF04] show a noticeable decrease of the ANN learning problem.

The size overhead in terms of dependability is measured against the overhead of the centralized approaches MTDC & AMAS. The minimum size overhead of a controller or broker is defined by $O_{CB,\min} = \frac{1}{2}$. Any overhead of an approach smaller than $O_{CB,\min}$ favors the challenger. A ranking of the five approaches is given in equation (2.23), placing O_{AHS} first and O_{AV} last, based on the overhead comparison made hand to hand.

The reliability gain is achieved by several factors. The most obvious one is the failing core tolerance F_t of each approach, but also the substitutability of the centralized cores. The less depending upon centralized cores an approach is, the higher the reliability and the increase of the failing core tolerance F_t . According to equation (2.11), AHS can tolerate the highest number of core failures, because of the flexible redundancy² achieved by the generalized core approach versus the fixed redundancy of the MTDC & AMAS. AHS fails only, if all N cores fail, but failing of all centralized decision units ($N_C \leq N_f$) leads to a complete failure of the centralized distribution mechanism. This is proven in [BBP13], stating that even with 20% brokers or controllers AHS outperforms the MTDC & AMAS in reliability, while AHS also has less overhead in size. As a consequence, the decentralized distribution mechanism keeps at least partial chip functionality and regains its distribution behavior, if for example failed cores are replaced or new cores added [vRBP11a]. AHS is closely followed by ANN, but not met, in terms of the highest F_t and the loss of the global learning unit is crucial to the self-control properties of ANN.

Another advantage of AHS is the symmetry of the cores, each one plays an identical role. The ANN comes close to AHS in regard of the symmetry, but already in [Mea89] is stated that ANNs do scale only hardly, if at all. AHS affects the task distribution and cores equally, which ensures the scalability of AHS onto System-on-Chip (SoC) and other sizing grids [BBP13]. For the MTDC and AMAS the scaling is simple, new working cores need to be noticed by the controllers and new controllers need to interact with all controllers shortly to place themselves within the controller hierarchy. Anyone using analog voter as a reliable architecture is aware that this architecture does not scale as well [NAN08, AN11].

Last, the capability to handle mixed-signal tasks and being able to migrate them between cores is stated for each approach. Only AV does not enable a mixed-signal task migration.

²The definition of flexible and fixed redundancy is given in [BBP13].

Table 2.3: Summary of the Comparison Results

	MTDC & AMAS		AHS	AV	ANN
Mechanism	centralized asymmetric	partly decentralized asymmetric	decentralized symmetric	centralized asymmetric	semi decentralized symmetric
Self-control					
• Self-configuration	state-evaluation	auctions	hormone loops	not feasible	learning period
• Self-optimization				voting	
• Self-healing	health values			not feasible	
• Self-reliance	global monitoring		local monitoring		global monitoring
Size overhead	$O_{CB,\min} = \frac{N_f+1}{2N_f+2} = \frac{1}{2}$		$O_{AHS} \prec O_{ANN}$	$O_{CB} \preceq O_{AV}$	$O_{ANN} \prec O_{CB}$
Real-time bounds to assign m tasks	$\mathcal{O}(m)$		$\mathcal{O}(m)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
Reliability Gain	$F_{t,CB,\max} = \frac{1}{2}$		$F_{t,AHS,\max} = 1$	$F_{t,AV} = 0$	$F_{t,CB,\max} = 1$
Scalability	yes		yes	no	no
Mixed-signal task migration	yes		yes	no	yes

2.5 Contributions

With the objective to develop a new analog and reliable architecture, which is robust and real-time capable, this thesis presents a highly reliable and dependable architecture synthesized from specification to layout. The contribution of all the necessary steps are outlined as follows:

1. Defining a reliable architecture,
2. Symbolic modeling of the reliable architecture,
3. Feasibility analysis of the model,
4. Failure analysis of the architecture,
5. Design of the components and monitors,
6. Evaluation, Fabrication and measurements of a silicon prototype.

In order to generate specifications describing a feasible, complex analog circuitry operating as a reliable task distribution system/architecture, the system has to be symbolically formalized in detail. Differential Algebraic Equations (DAEs) represent the functionality of each component. Assembling those DAEs, the system is described by symbolic functions. Further, sets of inequalities mark the constraints, which need to be applied with regard to timing issues, feasibility and robustness. Those sets are defined for each state the system can acquire. Combining the DAEs and the sets of inequalities into a huge equation system provides the formalized equivalent of the analog architecture. A symbolic analysis of the equation system derives constraints and dependencies for the specification.

The function of the reliable architecture is completed through the designed monitoring circuits. They allow to identify the critical dysfunctions within the decision modules and their associated cores and are able to control self-reliantly their eagerness to allocate and keep tasks. Only the sensor data is given, while the system self-configures itself and self-optimizes and self-heals, if a malfunction is monitored.

The decentralized approach of distributing tasks, simplifies the design of a mixed-signal sensor/actuator architecture. Each core decides only for itself, through its suitability and eagerness to allocate a task and its decision unit reacts accordingly. Tasks can switch between analog cores, yet digital cores can be addressed as well. A task switch from an analog PID controller to the digital PID controller and back is shown, marking the introduced architecture clearly as mixed-signal capable.

The evaluation and the measurements show the feasibility and functionality of the designed reliable architecture. It is validated that only satisfying system states are reached in a stable manner and all tasks successfully allocated by cores and reallocated, after a failing core drops the tasks. Furthermore, the timing constraints and power consumption are measured, while an allocation coverage is determined. Concluding, the proposed architecture is dependable and reliable.

3

Reliable, Mixed-Signal Architecture

Considering the advantages and disadvantages stated in Chapter 2 of the different approaches to distribution tasks within a system points to the need to identify, which criteria are important for an analog distribution system. Table 2.3 summarizes the criteria, which support this thesis.

The analog domain brings along a set of difficulties, for example a continuous state space, but also simplifies the design process, for example no need for a clock signal or guarding synchronicity. Further, analog cores are hard to generalize, tasks are in general very specific with the need of specially designed cores to meet the distinct specifications. Introducing further distinct roles, cores and controllers, would increase the overall complexity and worsen reliability. Further, those complex controllers were fully designed from scratch with 278 345 NAND-gates [BBP13]. Self-explanatory, this applies to brokers and the auction-based distribution equally. The less complex analog designs are, the more reliable those can be designed and can undergo more heavily determined testing to ensure robustness.

Distributing tasks based on locally done comparisons interacting with all participating cores through global loops is highly applicable for the analog domain. The global loops allow all participating cores to notice task drops, caused by any kind of failure. Therefore, monitoring instances can be placed at the cores and

need to be monitoring only locally, since no global instances exist. Also, the trigger functions allow to handle noisy signal effects easily, while they are still monitored to enable countermeasures, if the noise gets too heavy.

Since the analog components are designed from scratch, they can be minimized in size. This leads to the assumption that equations (2.17) and (2.27) also apply for an analog approach. Hence, no need to design complex centralized distribution units, taking also advantage of the symmetric distribution of the cores and its local decision units. The approach is assumed to be scalable and highly applicable for SoCs and other architectures. Lastly, the possibility to migrate suitable tasks for both domains between analog and digital cores represents a major gain in usability.

3.1 Artificial Hormone System with Analog Components

Designing digital circuits including analog component is a broad field of research. Approaches for an artificial hormone system with analog components (AHS-A) have recently been published [LPB⁺12, BvRHB13, vRSH⁺15]. The existing AHS architecture is expanded to address analog cores, which on their part reply and set the eagerness of the analog core for the different tasks [vRBBH12]. The decision unit and the communication of all the decision modules are still digital, but the local communication of a module with its core has to be converted for analog cores and vice versa, implying the need for an ADC and a DAC within the local AHS hardware as Figure 3.1 shows. An increase in complexity of AHS is the result [vRBBH12].

Equivalent to AHS, the amount of decision modules i corresponds to the number of task $i \in m$ the core applies to. The communication between the decision unit and the cores are in the simplest form 1-bit conversions to transport the *task_i on* signal one way and a *health signal*¹ the other. Though, in a more complex manner, the analog monitor connected to the analog core determines the *eager value* by monitoring for example degradation and environmental effects, while also taking the *task suitability* into account. Therefore, AHS-A is a mixed-signal architecture, using the digital AHS and integrating analog components. Most interference at

¹If core is healthy, the signal states true, else false.

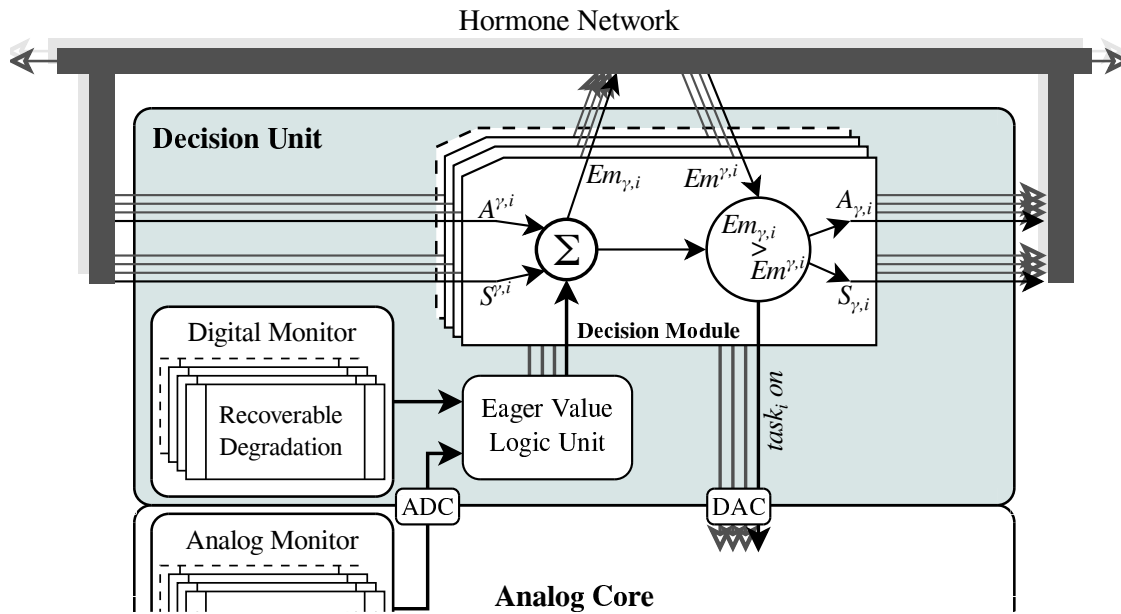


Figure 3.1: Digital Hormone Loop for Analog Cores (AHS-A) [vRBBH12, vRSH⁺15]

a core, like noise and process variation, is avoided due to the digital hormone system, but designing and verifying the cores using AHS-A is complex.

Figure 3.2 shows a simple example to indicate clearly the need of such mixed-signal approaches. A set of partly redundant cores is used to allocate three cooperating tasks to control an ABS braking system. A sensory *Wheel Signal Task* is allocated by a Digital Sensor Interface Core connected to the wheel sensor. The actuating *ABS Brake Task*, a software based task to avoid wheel blocking, is assigned by a RISC Core. It follows again a sensory task, the *Brake Control Task*, this time a hardware based task to monitor the hydraulic brake pressure, is allocated by an Analog Valve Driver Core connected to the braking system. A failure within any of those three tasks caused by the allocated cores leads, if monitored, to a reallocation of the according task to another suited core. The ABS system keeps functioning.

However, within this thesis this digital-based AHS is not followed any further.

3.2 Analog Artificial Hormone System

The Analog Artificial Hormone System (AAHS) is a mixed-signal, reliable architecture to ensure the reliable distribution of tasks within a redundant multi-core system. Each core is expanded by a decision unit, which is connected via the hor-

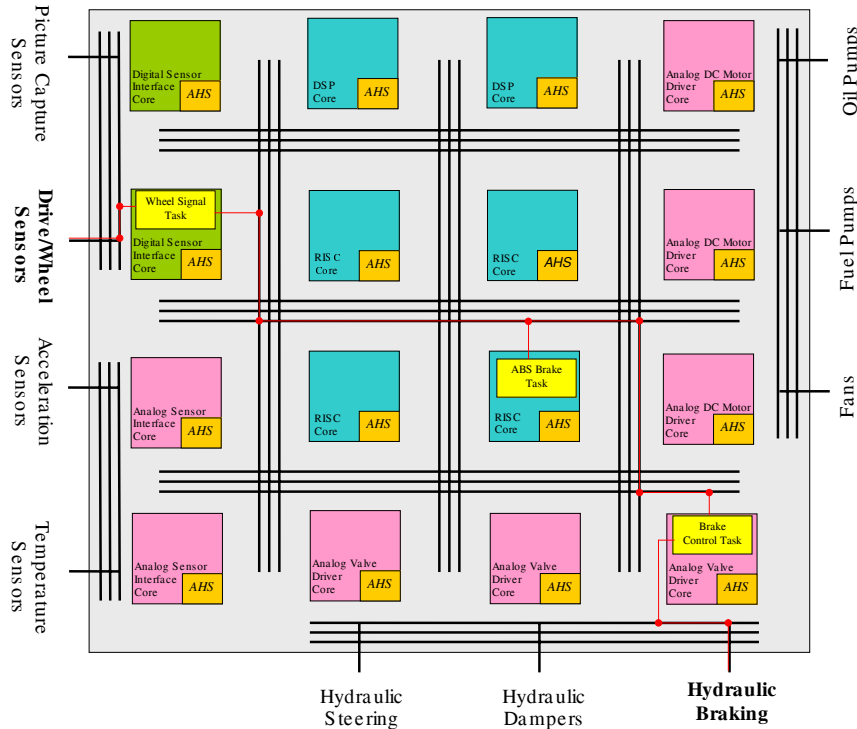


Figure 3.2: ABS Braking System using the AHS/AHS-A Architecture [vRSH⁺15]

more bus to the decision units of other cores, while each core itself is connected to the data bus also. Within each decision unit the number of decision modules symbolizes the number of tasks, the core applies to. Figure 3.3 illustrates a multi-core system using AAHS as architecture. The decision unit is described in the following, while the communication logic is presented in Chapter 3.2.3.

The generalized core and task concept of the AHS has to be redefined to be transferred into the analog domain. The five semi-specialized cores are replaced by only two core concepts forming the multi-core architecture:

Analog Cores: These cores are specified by the field of application the system is operating in, as for example a filter of acoustic signals or an active motor controller. Interface cores, interacting with the environment, fall into this category, as do so called analog special purpose cores. Thus, they cover every case an analog core can be designed for.

Digital Cores: Equivalent to the analog cores, this cores are specified by the field of application the system is operating in, for example interacting with the environment or fulfilling special purposes.

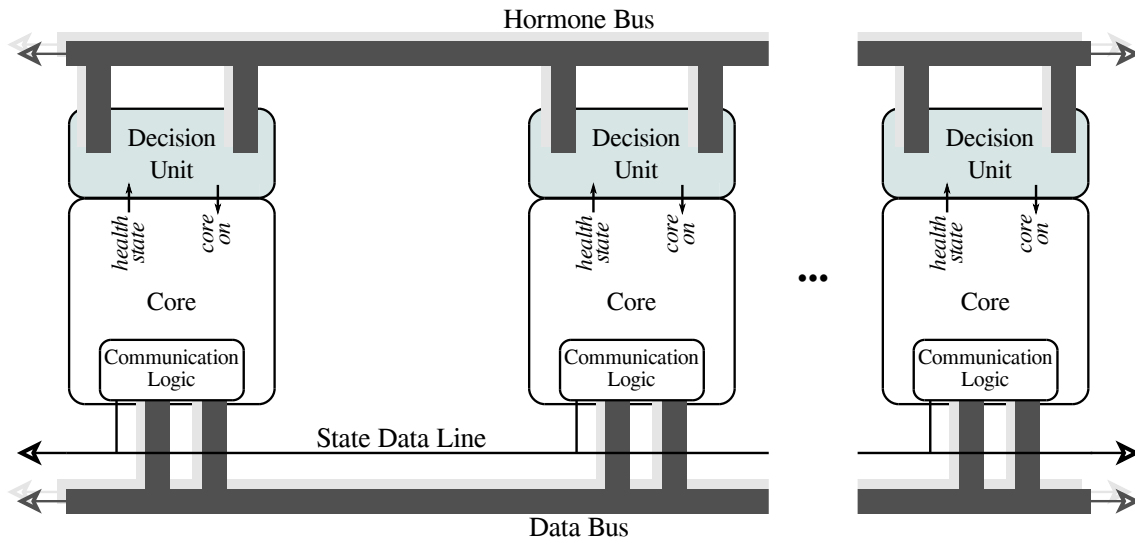


Figure 3.3: Model of a Multi-Core System using the Analog Artificial Hormone System

A decision module is attached to every core and task. Each core decides only for himself, if and which task is allocated based on the Global Hormone Levels of all tasks. Three analog hormone values exist:

Eager Value: The eager value represents the level of competence regarding a specific task, the suitability of the core to execute that task. The eager value is added locally to the Global Hormone Level.

Suppressor: The suppressor hormone suppresses the Global Hormone Level and counteracts the eagerness of a core to take a task. The suppressor is subtracted from the current Global Hormone Level, leading to a new Global Hormone Level.

Local Accelerator: The local accelerator counteracts the Global Hormone Level suppression to keep an allocated task running at the core. Without the local acceleration at the core the global suppression would lead to the immediate drop of the taken task. The local accelerator is also added locally to the Global Hormone Level.

To determine the extent of the hormones, two different levels are of importance to specify, if an allocation process is issued:

Global Hormone Level: The Global Hormone Level of a task is known to all cores applying to it. A stable, balanced hormone level is reached, if the task is allocated at one core only. A second scenario with all cores broken

down also results in a balanced hormone level. In any other case (failures or any other mischief) the hormone level is imbalanced, stimulating the cores to re-initialize the task allocation process.

Local Hormone Level: The Local Hormone Level of a task is calculated locally at each core and known exclusively. It may differ significantly to one another. A stability condition is not existing. The level is balanced, if the Global Hormone Level is stabilized, only affected by a change of the own eager value.

Figure 3.4 illustrates the control loop of the AAHS, showing the interaction of the three analog hormones. For each task the decision unit applies to, a decision module is needed. Hence, the amount of modules is scalable, according to $i \in m$ tasks the core applies to. The Latin letters such as i as indices indicate a task $T_i : i \in \{1..m\}$. To distinguish between the cores Greek letters such as $C_\gamma : \gamma \in \{1..N\}$ are used as indices. Contrary to AHS, a distinction between sent and received hormones is not done. A hormone of any type carries subscripted indices either

- only in regard to task T_i or
- only in regard to core C_γ or
- in regard to both T_i and C_γ .

Global accelerators are not intended so far. Some ideas, however, of how to implement and use them are presented at Chapter 7.2. At each core, respectively at each decision module, the Global Hormone Level is added to the own eager value. This modified value is held against the threshold value of a trigger. Only if the threshold is met, the core allocates the task and keeps it until the negative threshold is hit. The continuous manner of the analog domain ensures that the core with the highest eager value triggers the first. A cyclic evaluation of the hormone level is henceforth obsolete, the hormone evaluation happens by continuous time operation.

Hence, the real-time bounds are depending on the slew rates of the amplifiers and the signal traveling times due to parasitic resistors and capacitors, within the system.² Assigning m tasks is done in a continuous manner, contrary to the cycle based allocation routine.

²Yet, the results in Chapter 6 show that the traveling times could be neglected in comparison to slew rates and bandwidth of the active analog components.

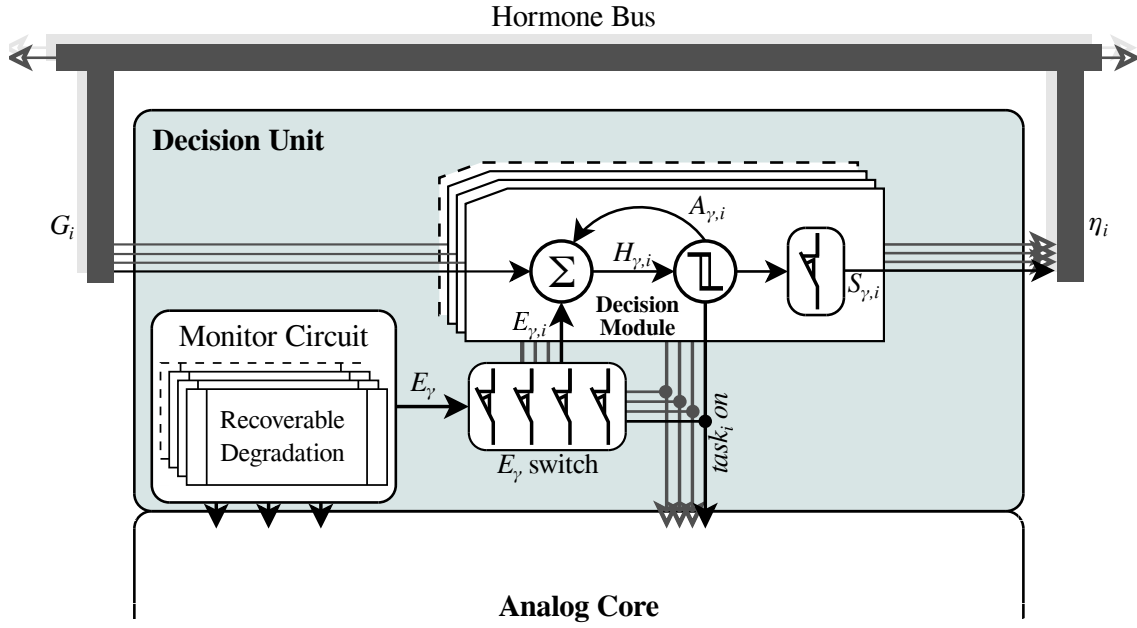


Figure 3.4: Analog Artificial Hormone Based Control Loop (AAHS) [vRSH⁺15, Ch. 4]

3.2.1 Defining the Analog Hormone System

The definition of the hormone loop, as it is stated by equation (2.38) for AHS, is not as easily done for AAHS. Twice the maximum communication time of any two cores guarantees for a cyclic approach that all cores received the hormones and answered accordingly. For an analog approach the hormone loop depends upon the time the Global Hormone Level adjusts to the new value, which can be described by a simple RC-signal modeled by the first order differential equation

$$G_i + \tau_{G,i} \cdot \dot{G}_i = \eta_i : i \in m \text{ tasks}, \quad (3.1)$$

where G_i is the signal of the Global Hormone Level for task T_i and $\tau_{G,i}$ represents the time to adjust to the new Global Hormone Level η_i . Each task is represented by its own hormone loop, since $G_i : i \in \{1..m\}$ applies.

The cores constantly receive G_i and base their decisions, whether to allocate or reject the task T_i , upon it. Yet, $\tau_{G,i}$ seconds after an allocation occurred, the new stable hormone level η_i is reached. Hence, the time $\tau_{G,i}$ to adjust to η_i corresponds to a single, digital hormone cycle³ of AHS, as seen in Figure 3.5(a). Rejecting the task has no impact upon G_i . Allocating the task T_i , however, leads to two reactions:

³The digital hormone cycles are described and defined in detail in [vRBP11b].

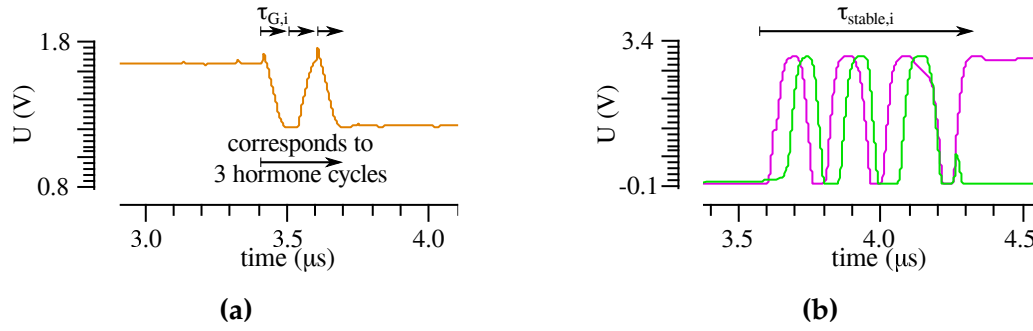


Figure 3.5: Sketches outlining the difference of (a) $\tau_{G,i}$ and (b) $\tau_{stable,i}$ with randomly chosen scale line on the time axis

1. The suppressor signals $S_{\gamma,i}$ are sent, resulting in a new Global Hormone Level $\eta_i = G_i - \sum_{\gamma=1}^N S_{\gamma,i}$.
2. The local accelerator $A_{\gamma,i}$ is activated, adjusting to its new value $\lambda_{\gamma,i}$ by

$$A_{\gamma,i} + \tau_{L,\gamma,i} \cdot \dot{A}_i = \lambda_{\gamma,i} : i \in m, \gamma \in N. \quad (3.2)$$

Noticeable, the local accelerator needs to apply before the new Global Hormone Level η_i is reached. The different time constants for those loops can be specified exactly and the components are designed accordingly during the analog design process. If several cores allocate the task T_i , the Global Hormone Level drops at least twice the suppressor value. This enforces a task drop of task T_i at every core. Such multiple (≥ 2) allocations can only apply, if allocation decisions are done within $\tau_{G,i}$ seconds, while the Global Hormone Level is unstable. The allocation decision triggers at each core for each applying task, if and only if equation (3.3) is true. The equation part $G_i + E_\gamma + A_{\gamma,i}$ corresponds to the Local Hormone Level.

$$\theta_{\gamma,i} \leq G_i + E_\gamma + A_{\gamma,i} \quad (3.3)$$

Without the global accelerator the stability constraint (equation (1.5) on page 28) of the AHS hormone loop is automatically ensured for AAHS. However, the true stability constraint for AAHS is specified by the time it takes to have task T_i allocated once and G_i being stabilized again:

$$\tau_{stable,i} = \tau_{G,i} + \operatorname{argmin} (\forall_{\xi>0} G_i(\tau_{G,i} + \xi) = G_i(\tau_{G,i})) : i \in m. \quad (3.4)$$

Figure 3.5(b) sketches the circuit behavior, which defines $\tau_{\text{stable},i}$. Further, equation (3.4) also states, if the specified real-time bounds can be fulfilled, since $\tau_{\text{stable},i}$ is of importance for the WCTDT calculation. Each decision unit of the cores decides continuously upon taking a task. Due to this continuous character of AAHS all tasks are allocated by cores simultaneously in time, only slowed down by the according $\tau_{\text{stable},i}$:

$$\text{WCTDT}_{\text{AAHS}} = \tau_{\text{AAHS}} = \sum_{i=1}^m \tau_{\text{stable},i} \text{ seconds.} \quad (3.5)$$

$$\Rightarrow \mathcal{O}(m) \quad (3.6)$$

It follows that in terms of complexity of the real-time bounds to assign m tasks AHS and AAHS are equivalent. The true comparison has to be done by measuring the real timings.

Table 3.1: Size Increase of the Analog Artificial Hormone System

\square_{AAHS} : Size of the distribution mechanism, which is added to every core

$$\Rightarrow \text{Percentage increase of each core } \square_{\% \text{AAHS}} = \frac{\sum_{i=1}^N (\square_{\text{AAHS}})}{N}$$

Equivalent to AHS, the size of the decision modules of AAHS is mapped by \square_{AAHS} as Table 3.1 states. Again, the size of the monitor circuits and the global routing is neglected. The equivalence of the definition of \square_{AHS} and \square_{AAHS} results from the equal running conditions (equation (3.7)), due to the flexible redundancy and the decentralized approach.

$$\text{running}_{\text{AAHS}} : N > N_f \quad (3.7)$$

This leads to the definition of the size overhead of AAHS by equation (3.8). Further, it is to be assumed that equation (2.17) from page 39 also applies to the analog approach of a hormone system (equation (3.9)).

$$O_{\text{AAHS}} = \square_{\% \text{AAHS}} \quad (3.8)$$

$$\square_{\% \text{AAHS}} < O_{\text{CB},\text{min}} = \frac{1}{2} \quad (3.9)$$

Table 3.2: Preliminary Summary of the Analog Hormone System

	AAHS
Mechanism	decentralized symmetric
Self-control	hormone loops
• Self-configuration	
• Self-optimization	
• Self-healing	
• Self-reliance	local monitoring
Size overhead	if $O_{AAHS} < 50\%$ than $O_{AAHS} < O_{CB}$
Real-time bounds to assign m tasks	$\mathcal{O}(m)$
Reliability gain	
Scalability	
Mixed-signal task migration	yes

Since equation (3.9) is assumed to hold, equation (2.27) also applies to AAHS, concluding equation (3.10).

$$O_{AHS} \equiv O_{AAHS} \tag{3.10}$$

$$\square_{\%AAHS} \stackrel{?}{\leq} \square_{\%AHS} \tag{3.11}$$

Therefore, an overhead comparison in terms of dependability between AHS and AAHS is unrewarding. An accurate comparison of the size of the modules (in μm^2) determines, if AAHS precedes AHS in overhead (equation(3.11)).

The mentioned characteristics are measured/compared against the columns of Table 2.3 on page 50, which serve as benchmarks. A first summarizing peek of the AAHS is given in Table 3.2. The gray colored cells are already defined by choosing AHS as reliable architecture enhancing it to AAHS. The filled white cells have just been defined, while the white and empty cells still need to be characterized. The subscript CB nevertheless represents the centralized controller and the auction-based distribution system. Further, it has to be stated that

- the self-control property ensures the dependability and robustness of AAHS,

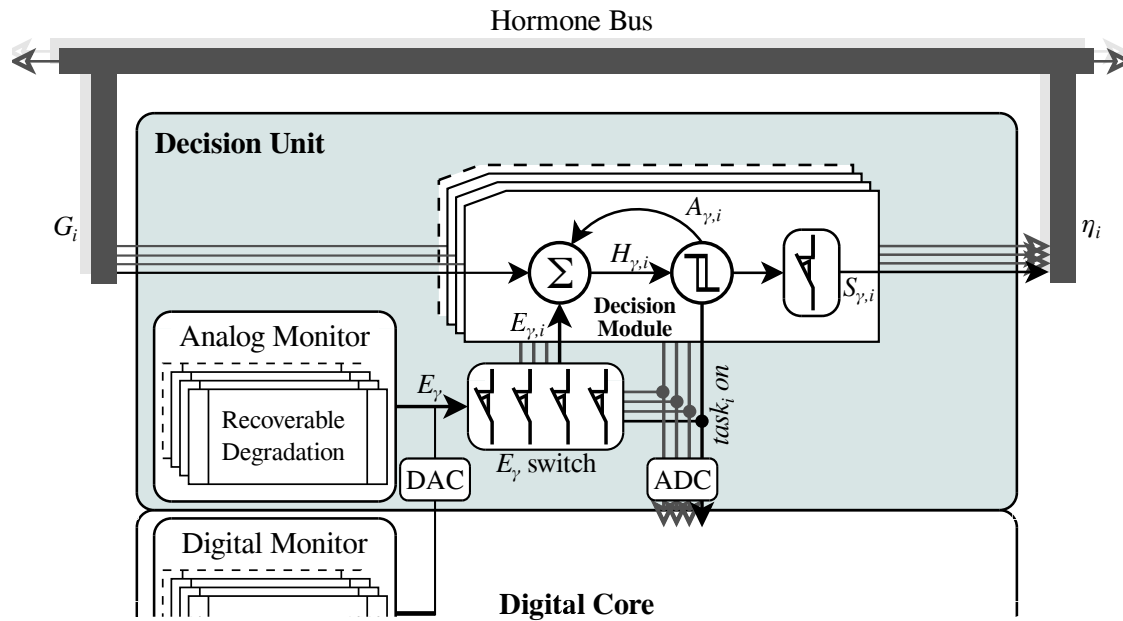


Figure 3.6: Analog Hormone Loop for Digital Cores

- any task has to be allocated within its real-time bounds,
- no double allocations may occur, but a task dropped by a core has to be reallocated by another,
- the design must be scalable for any N cores and any $m < N$ tasks.

The self-reliance heavily depends upon the quality of the monitoring. Local monitors can be simplified, since their task area is narrowed down to the specific characteristics of their respective cores.

The last entry of Table 3.2 states that AAHS, equivalent to AHS, is mixed-signal task migration capable. Figure 3.6 shows the most basic single signal line connectivity of a processing core to the analog hormone system. The rudimentary connection between the core and AAHS allows a *task_i on/off* and a *core health state* signal only. Though, the communication between the core and the decision modules can be advanced.

All of the mentioned criteria favor the design of an Analog Artificial Hormone System (AAHS), even though no proven predictions can be made for the reliability gain and scalability until now.

3.2.2 Designing the Analog Hormone System

Descending now from the algorithm layer of the functional domain to the macro layer of the structural domain, see Figure 1.1 at page 3, implies the need of a fundamental design decision. Since the three analog hormones are represented by signals on wires, the analog domain demands the decision of whether the voltages or currents of the electrical signals carry the hormone information. The structure of AAHS has to be designed according to this decision.

Being done with the definition of AAHS, the hierarchically top-down design methodology splits into two paths now, one able to process the voltage-based and the other to process the current-based information, which are presented here:

1. Defining the analog hormone system
2. Designing the analog hormone system
 - The decision module
 - The Schmitt Trigger
 - The voltage-based module
 - The current-based module
 - The hormone bus
 - The voltage-based bus
 - The current-based bus
 - The E_γ Switches
 - The voltage-based E_γ Switch
 - The current-based E_γ Switch
 - The monitor circuits
3. Synthesizing, layouting and fabricating the two analog hormone systems
4. Evaluation of schematics, the extracted view and the measurements of the silicon

The complete design process including the third and fourth design step are done in Chapter 5. Noticeable is that the monitor circuits need to provide either an eager value as current or as voltage, apart from that they are freely designable. Figure 3.4, illustrating the top layer functional description of AAHS, also indicates the top-down design process with the design of the modules, the hormone buses and the monitors.

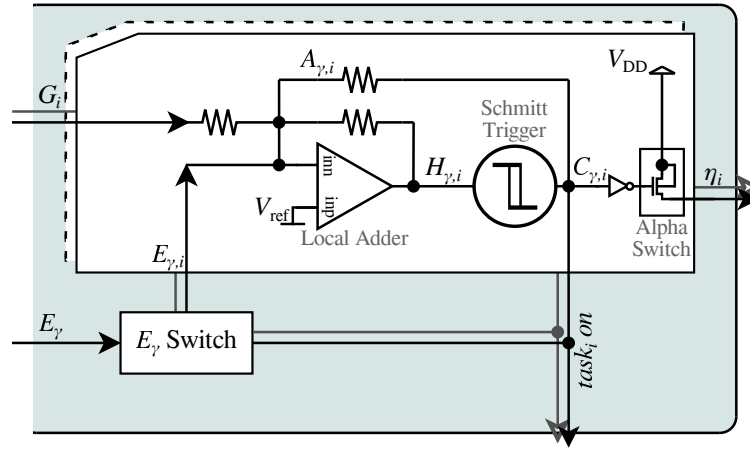


Figure 3.7: Implementation of the Decision Module with OpAmps [vRSH⁺15, Ch. 4]

3.2.2.1 The Decision Module

The Schmitt Trigger The Schmitt Trigger represents the centerpiece of the decision unit. Its decisions affect the full system extensively. The input signal is constantly varying, representing the Local Hormone Level $H_{\gamma,i}$. If $H_{\gamma,i}$ meets the threshold θ the digital output signal $C_{\gamma,i} \in \{\mathbf{t}, \mathbf{f}\}$ changes, indicating a task allocation or a task drop. Hence, the Schmitt Trigger is subject to degradation effects, especially if it is realized as an OpAmp applying the positive feedback to the non-inverting input. Typical effects are threshold voltage drifts [vRSH⁺15] and increasing offsets, caused by NBTI and HCI stress.

The Voltage-Based Module The voltage-based architecture uses OpAmps as base components to control the different levels of hormone values. Figure 3.7 illustrates a sketched schematic of the basic voltage-based implementation. The decision module is realized with an OpAmp as the Local Adder. The resistors represent the various ratios the hormones have. The Local Hormone Level $H_{\gamma,i}$ is the sum of all hormone values, the output of the Local Adder, as well as the input of the Schmitt Trigger. Following the Schmitt Trigger, a basic CMOS-Inverter is needed to operate the Alpha Switch (the boxed single transistor), which is needed for the suppressor hormone $S_{\gamma,i}$. A second decision module is indicated, stating that the presented implementation of the decision unit is applying to two tasks.

Equation (3.3) has to apply for all tasks, leading to equation (3.12):

$$\forall i \in \{1..m\}, \exists \gamma \in \{1..N\} : \theta_{\gamma,i}(V) \leq G_i(V) + E_\gamma(V) + A_{\gamma,i}(V) \quad (3.12)$$

Whenever, in a continuous manner, the equation (3.12) is fulfilled for task T_i at core $C_\kappa \in \{1..N\}$, the decision unit triggers and enables the core to execute task T_i . Once a task is taken by core C_κ , the core has to suppress the Global Hormone Level $G_i(V)$ to its new value $\eta_i(V)$. Further, the local accelerator $A_{\kappa,i}(V)$ is activated to hinder the fulfillment of

$$-\theta_{\kappa,i}(V) \leq \eta_i(V) + E_\kappa(V) + A_{\kappa,i}(V), \quad (3.13)$$

otherwise C_κ drops the allocated task T_i immediately, causing an oscillation.

The input of an AAHS decision module contains the Global Hormone Level, its eager value and $V_{\text{ref}} = \frac{V_{\text{DD}} + V_{\text{SS}}}{2}$. While it has two output signals, one is the suppressor connected to the hormone bus, the other is the *task_i on* signal, which is either **t** or **f**. The output of the suppressor value $S_{\gamma,i}(V)$ is aligned in respect to V_{ref} to be either $V_{\text{ref}} - S_{\gamma,i}(V)$ or V_{ref} , since each OpAmp is aligned to V_{ref} , the zero point of the AAHS. The eager values are provided as voltages by the monitor circuits, which guard the cores. The values range from V_{ref} , the zero point, to V_{DD} .

The Current-Based Module Alternatively, a current base architecture to control the different hormone values is implemented. Operational-Transconductance-Amplifiers (OTA) are used as signal processing units. The sketched OTA implementation is shown at Figure 3.8. The decision module is realized with two OTAs, the first is the Measure OTA, the second the Res. OTA. The output of the Measure OTA determines the Global Hormone Level G_i and thus the suppressor value as current. The three currents $S_{\gamma,i}$, E_γ and $A_{\gamma,i}$ are added and due to the Res. OTA converted into the Local Hormone Level $H_{\gamma,i}$. The output of the Schmitt Trigger is current mirrored into the local accelerator and the global suppressor hormone, and CMOS inverted to indicate the *task_i on* signal. The second decision module is indicated, stating that the presented implementation of the decision unit is applying to two tasks. The eager values itself are currents, provided by the monitor circuits of the cores.

E_γ Switches If any core is applying to several tasks, an E_γ Switch is needed for the several decision modules. The switching circuit is a simple exclusive or circuitry, only one task can be allocated by a core. If a core is occupied, the eager value needs to be blocked for the other decision modules the core has, except the one which triggered the allocation. The eager values pass through again as soon

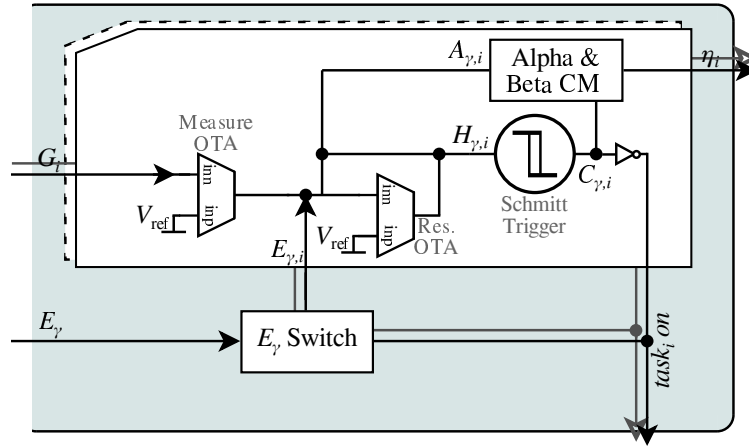


Figure 3.8: Implementation of the Decision Module with OTAs [vRSH⁺15, Ch. 4]

as the core drops its task. Therefore, for each decision module an $E_{\gamma,i}$ is defined according to equation (3.14).

$$E_{\gamma,i} = \begin{cases} E_{\gamma} & \text{if } (\forall_k C_{\gamma,k} = \mathbf{f}) \vee (C_{\gamma,i} = \mathbf{t}), \\ 0 & \text{if } (\exists_k C_{\gamma,k \neq i} = \mathbf{t}) \wedge (C_{\gamma,i} = \mathbf{f}). \end{cases} \quad (3.14)$$

Two distinct types of eager values exist, one voltage-based and the other current-based, being able to influence the Local Hormone Level $H_{\gamma,i}$ unfiltered and immediately. For the voltage-based eager value, the switch is configured as follows:

- If core C_{γ} has not allocated a task so far, the eager values $E_{\gamma,i}(V) > V_{\text{ref}}$ and influence the value of each trigger function $\theta_{\gamma,i}(V)$.
- If core C_{γ} has allocated task T_i , the eager value $E_{\gamma,i}(V) > V_{\text{ref}}$ continues to apply, but all other $E_{\gamma,k \neq i}(V) = V_{\text{ref}} : k \in m$.
- A capacity at the gate of one switching transistor pair (one transistor to cut out the eager value, while the other than connects to V_{ref}) implies implicitly an order of importance of the tasks, which task to take first.⁴

Equivalent, a current-based E_{γ} Switch is configured as follows:

- If core C_{γ} has not allocated a task so far, the eager values $E_{\gamma,i}(A) > 0$.

⁴The actual effect of the capacity to be seen is a decrease of the slew rate of the signal $C_{\gamma,i} \in \{\mathbf{t}, \mathbf{f}\}$, which is controlling the switch. Figure 6.2(a) on page 137 shows the decreased slew rate of $C_{3,2}$ affected by the capacitor.

- If core C_γ has allocated task T_i , all other eager values $E_\gamma(A) = 0$ for the trigger functions $\theta_{\kappa \neq \gamma, i}(A)$.
- The order of importance of the tasks is given by the different sizes of the switching transistors.

However, a local current mirror needs to duplicate the eager value signal beforehand according to the number of tasks the core applies to. For simplicity, the general eager values E_γ are considered again, depending only on cores, not on the tasks as well. Yet, the distinctions of E_γ seen in equation (3.14) are regarded during the design analysis again (Chapter 5).

3.2.2.2 The Hormone Bus

Both types of architecture are affected by both Kirchhoff's laws. Kirchhoff's Current Law (KCL) allows for the current-based architecture to simply put the suppressor value upon the hormone bus. However, for the cores to measure the current of the Global Hormone Level, a shunt resistor is needed (Figure 3.10(a)). Since precise resistors are hardly realizable, the shunt resistor is replaced by an OTA designed as resistor (Shunt OTA). Each core measures, using the Measure OTA, the potential difference between the negative input and the output of the Shunt OTA to determine the Global Hormone Level locally for the decision module.

Contrary, Kirchhoff's Voltage Law (KVL) implies that the sum of the potential difference of each closed mesh within an electric network equals zero. Voltages do not add up by simply attaching the signal lines, but by a voltage adder realized with OpAmps, outputting the Global Hormone Level (Figure 3.9(a)). Since all cores are parallelly attached to the hormone bus, the potential difference at each core input is equivalent.

The Global Adder and the Shunt OTA for each task display a single point of failure, but for that specific task only. Those single points of failure can be eliminated, as the two following two-fold bus structures show:

1. Regarding the Global Adder of the voltage-based architecture as task, the task can be doubled and the AAHS be used to distribute the task between those two adder cores. The adder input signals and the single line adder output signal are regarded as the data lines. Now, a special feature concerning only two decision modules of AAHS comes in handy. Connecting the

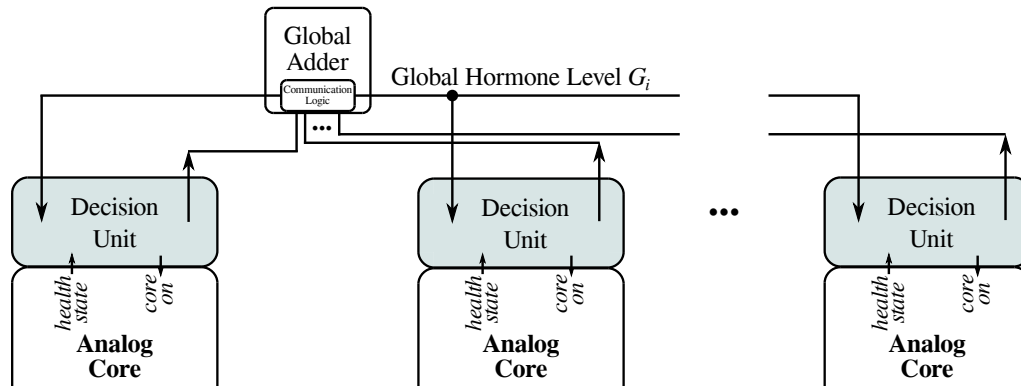
two modules so that the suppressor of one module is connected to the input of the other and vice versa eliminates the Global Adder as single point of failure for this task. Each adder core connects itself to the data lines to receive the adder input signals. The output signals are kept separate as redundant Global Hormone Level lines, also excluding them as single points of failure. Only one core is active (since it suppresses the other) and executes the adder task. A two-fold redundant hormone bus is the case, as shown in Figure 3.9(b).

- For each redundant hormone bus line of task T_i the according Local Adder at the decision module needs additional input pins connected to the redundant lines.
- The bus lines are either aligned to V_{ref} , indicating that the line is unused or the task is not allocated, or are transmitting the suppressing hormones. In either case the Local Adder is evaluating a correct value for the Local Hormone Level $H_{\gamma,i}$.

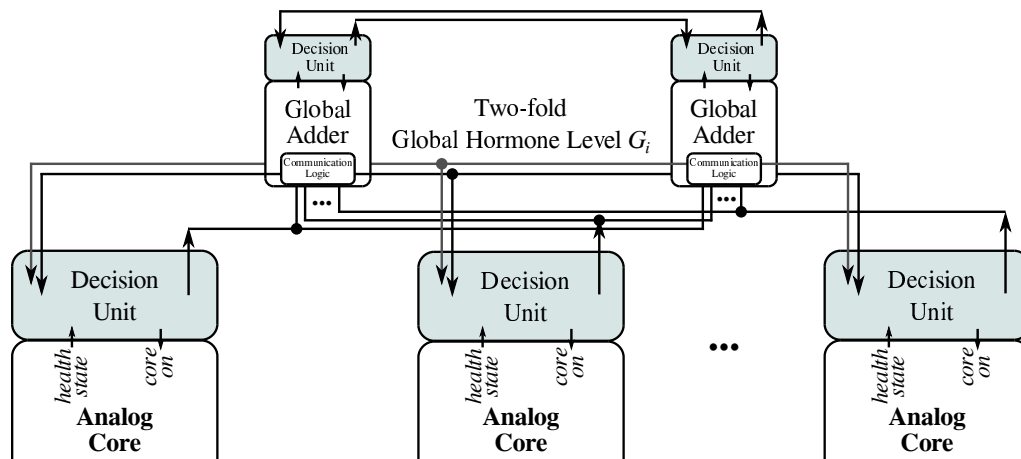
To ensure the functionality of the adder core, monitor circuits are guarding the adder cores, checking for example the difference of the positive and negative input of the OpAmps or using a *heartbeat signal* for the wires. Chapter 5.4 focuses on the different monitor circuits.

2. Achieving a redundant hormone bus implies for each task to have more than one line usable as Global Hormone Level. Again, two-fold redundancy of the hormone bus is gained by doubling the Global Hormone Level line. Two cross connected decision modules control, which bus line is taken, while monitor circuits are guarding the lines (Figure 3.10(b)). Further, two circumstances enable the elimination of the global Shunt OTA as single point of failure. First, the negative input of the used OTA is connected to V_{ref} . Second, the potential difference of each wire attached parallel to the Global Hormone Level line is equivalent. Those two allow to duplicate the Shunt OTA into each decision module to measure the potential difference locally at each core. However, the two-fold redundancy can only be achieved through several editions to the decision module using OTAs:

- The suppressor at each core needs to be mirrored twice.
- The two suppressor outputs are connected to the two different Global Hormone Level lines.



(a) Hormone Bus with one Global Adder



(b) Redundant Analog Hormone Bus

Figure 3.9: Voltage-based Hormone Bus Structure

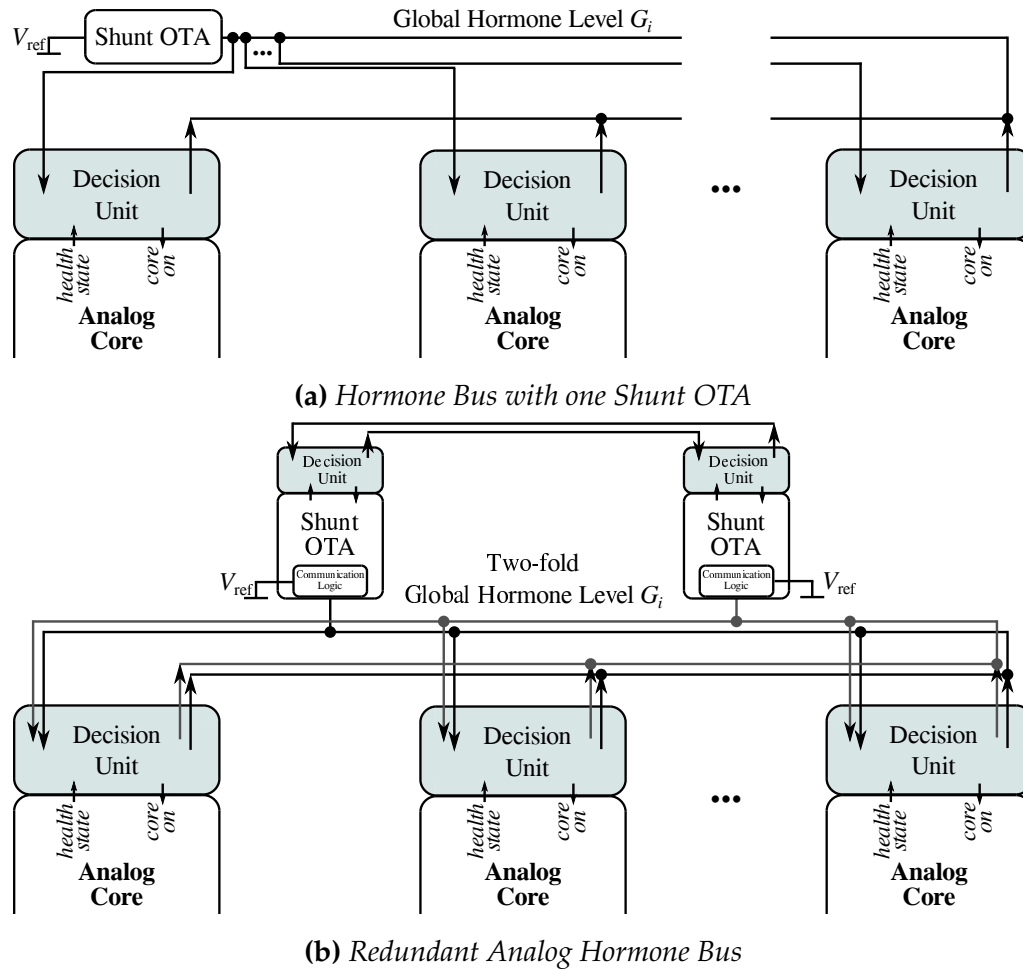


Figure 3.10: Current-based Hormone Bus Structure

- Only one bus line is active at all times, caused by the mutually suppression.
- The local Shunt OTA needs to be doubled also, connected to the new redundant Global Hormone Level line.
- The outputs of the local Shunt OTAs are connected to likewise doubled Measure OTAs.

All in all, such a two-fold redundant bus with local Shunt OTAs would double the size of the decision unit most likely. Figure 3.10(b) shows the implementation of a redundant hormone bus using the current-based architecture.

3.2.2.3 Comparing the Architectures

Comparing the two AAHS architectures, several advantages of the current-based implementation can be stated:

- OTAs can be used open-looped. The necessity of feedback loops for linear operation as needed for OpAmps does not exist.
- The internal design of OTAs is realized by transistors only, no need of compensation capacitors and resistors.
- Precise resistors are hardly realizable/feasible in integrated circuits favoring analog embedded systems built by OTAs.

Field Programmable Analog Arrays (FPAA) are using OTAs as base elements. Contrary, the voltage-based implementation with OpAmps benefits in the following:

- OpAmps are easier to be designed from scratch for prototypical development or a proof-of-concept.
- OpAmps score with low offset voltages and high accuracy, especially in regard of low frequencies.
- OpAmps can be operated with high impedance inputs and low impedance outputs.

Both types of architectures are evaluated at the end at Chapter 6 stating, if one architecture surpasses the other.

3.2.2.4 Open Design Measures

The implementation of analog circuits using several OpAmps or OTAs, resistors, current mirrors, capacities and single transistors implies a huge set of design and process parameter, which all need to be set. Determining the dependencies and ratios of the hormones is the most crucial step of the development of the AAHS. Only if a feasible region for the AAHS to operate in exists, the system can be designed. Following, the arisen design measures of AAHS:

1. Determine the dependencies and ratios of G_i , $S_{\gamma,i}$, $A_{\gamma,i}$, E_γ and $\theta_{\gamma,i}$ to operate the analog hormone system within a feasible region.
2. Defining the specification of all the different amplifiers.

3. Designing and sizing the transistors, current mirrors, resistors and capacitors.
4. Withstand process variation and other failure sources.

Chapter 4.1 focuses upon the dependencies of the different hormones. With the known dependencies the specifications can be derived for the OTAs and OpAmps. It is important to interpret the dependencies and ratios in respect to the specification needed for the amplifiers, which is done in Chapter 5.1 and followed by the synthesis, design and sizing of all components.

For example, different slew rates within the decision modules are needed to ensure the functionality. However, timing constraints of real-time bounds must be preserved, as well as the time $\tau_{G,i}$ to adjust the Global Hormone Level η_i (equation (3.1)). Any delays or triggered faulty decisions, while the Global Hormone Level was adjusting, acts contrary to $\tau_{\text{stable},i}$ (equation (3.4)) and may lead to a (non-ending) oscillation, violating any real-time bounds. The slew rates must be kept in relation towards one another, especially the global one must not be too slow and designed accordingly.

3.2.3 Task (Re-)Allocation Process

So far tasks were viewed from a general perspective, not clearly defined. However, to specify the allocation process a clear definition of tasks distributed by the AAHS is needed. The analog cores are specified by the field of application the system operates in. Each core has a purpose, a task, which is to be fulfilled/executed. Therefore, for this thesis the term *task* is defined as:

Definition 3.2.1 (Task)

Work has to be done, a purpose has to be fulfilled.

Only the core, which is doing the work, knows exactly what kind of work has to be done. Since any task distribution system is not interested in knowing what kind of task has to be executed, only the distribution and execution is of interest. The work, the tasks, are defined by the wide-range of different analog and digital cores:

- analog/digital controller (e.g. PID controller) controlling a motor,
- output stage/amplifier (e.g. specified for low power),
- analog/digital filter (e.g. finite/infinite impulse response (FIR/IIR) filter),

- analog/digital pulse-width modulation (PWM),
- analog-digital/digital-analog converter (e.g. Sigma-Delta modulators, Flash-ADC),
- charge pump,
- voltage adder,
- clock generator,
- and many other.

Some of the cores need to be connected to sensors for example to low pass filter an acoustic signal, others to actuators for example to steer a robotic arm, and still others provide new signals. In some cases the analog signals from the data bus need to be converted into the digital domain and afterwards converted back into analog signals. Obviously, the conversion errors have to be monitored as closely as the digital core itself for the eager values.

In the following chapters, the hardware of the communication logic, which is seen in Figure 3.3, is presented. Next to AAHS, every core needs the allocation hardware to connect itself to the data bus.

3.2.3.1 Allocating Tasks

The process of allocating task begins, if equation (3.3) is fulfilled. The decision unit triggers and the signal $C_{\gamma,i} \in \mathbb{B} : \mathbb{B} = \{\mathbf{t}, \mathbf{f}\}$ is set to \mathbf{t} , which corresponds to *task_i on*. The *task_i on* signal controls the connections to the data bus, activating only the connections according to the allocated task. This circuit component structure is called Communication Logic. In its most basic form, an analog communication logic consists of several basic transmission gates, which are connected to the single wired bus lines. The core receives the data, progresses with the execution of the task and forwards the output back to the data bus. The process is illustrated in Figure 3.11 and regarded as the standard procedure of the task allocation.

Next to the allocation process, the task transfer from one core to another core gives AAHS its flexibility and ensures the satisfaction of the self-control properties. Both terms *task reallocation* [vRSH⁺15] and *task migration* [vRSH⁺15] are indicating the same process of transferring a task between cores. They only differ in the cause that leads to the transfer, as stated in Definition 3.2.2 and 3.2.3.

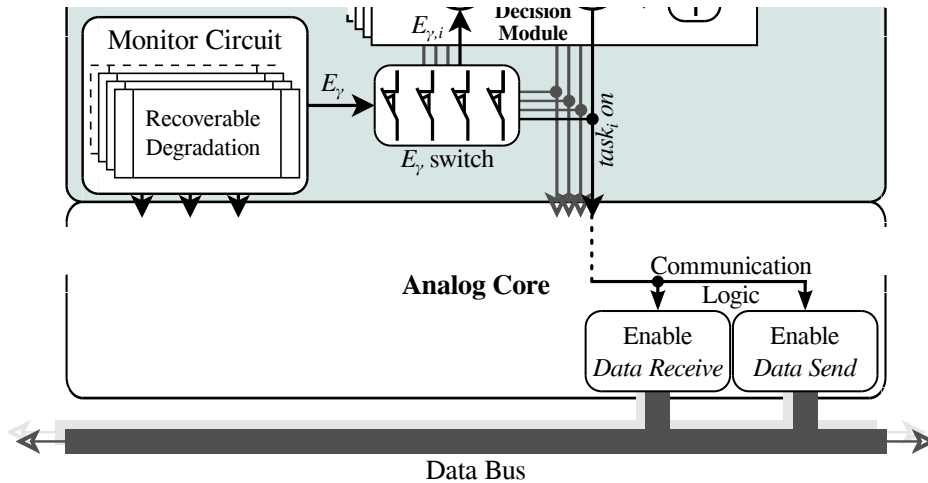


Figure 3.11: *Communication Logic of a Core*

Definition 3.2.2 (Task Reallocation)

The reallocation is a reaction to a severe failure, which led to a core outage including losing all data and a restart of the task.

Definition 3.2.3 (Task Migration)

The migration is a pro-active behavior, leading to a task transfer including data, initiated by monitor circuits checking on soft failures. The task state is picked up and execution continues from there on.

In [vRSH⁺15] a first approach of task migration is presented. The approach focuses on the different specification each core is designed with. Those indicate the different suitabilities to execute different tasks, especially throughout the two domains. Three distinctions to transfer tasks have to be made:

- A transfer between cores with similar specifications is viewed as the standard case. The only difference, which needs to be regarded, is the differing eager values of the cores. [vRSH⁺15, Ch. 4]
- To transfer tasks between cores with diverging specifications two cases have to be considered [vRSH⁺15, Ch. 4]:
 1. „A task transfer from the less suitable core to the better suitable core happens as easily as from cores with similar specifications.“ [vRSH⁺15, Ch. 4]
 2. Contrary, less suitable cores do either not allocate that task or the failing specifications can be neglected. The eager value indicates the level

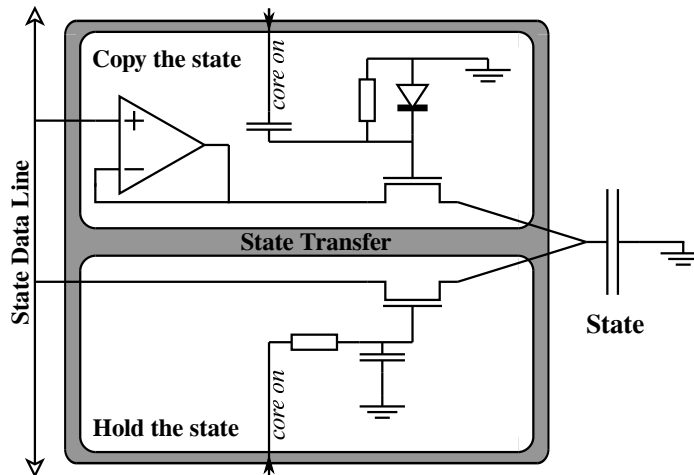


Figure 3.12: *Analog State Transfer*

of competence to execute that task. In this case the eager value is either zero or noticeably below the optimal value [vRSH⁺15, Ch. 4].

- The last case is the task transfer between different domains. Again, only the differing eager values handle the distribution process, independent of any specification and domain. Tasks, allocatable by both domains, are rare and highly specified and specialized [vRSH⁺15, Ch. 4]. A task transfer across domains is visualized in Figure 3.13 on page 77.

Chapter 5.3.3 shows the full system design needed to transfer tasks, which is used for the simulations.

3.2.3.2 Migrating within one Domain

The pro-active behavior of migrating tasks allows considering to transfer the state of the task, which is currently executed. To enable a transfer of states a new data line is needed, called *state data line*. The cores are connected to the state data line and pull any data while allocating the task, $task_i$ on is set to **t**. Soon afterwards the pull line is disabled, while the core pushes data continuously onto the state line. The state line with the pull and push wiring of a core is illustrated in Figure 3.12. The advantage of the continuously data pushing is the loss of the distinction between reallocation and migration, for the price of constantly occupying a data line. A state transfer in the analog domain is only applicable for capacities within a component. For example the integral part of an analog PID controller, which is

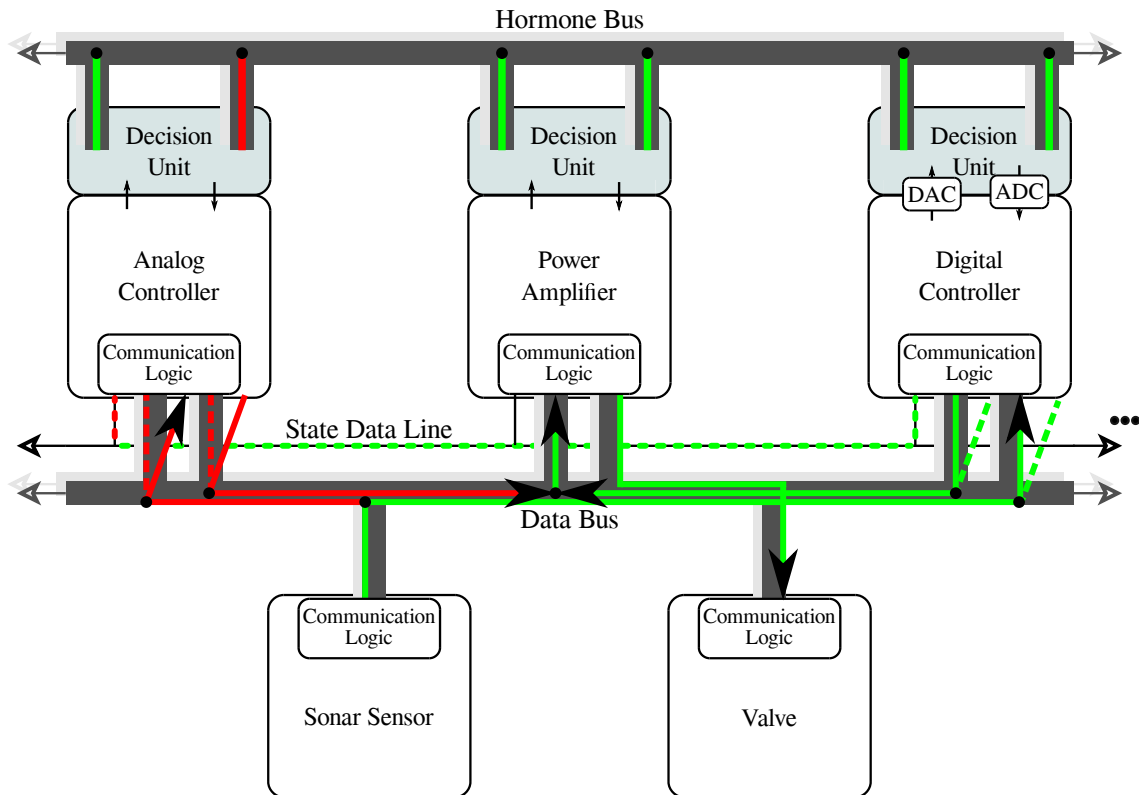


Figure 3.13: Task Migration between an Analog and a Digital Core

represented by a capacity, can be transferred, if possible. This also applies for low-pass filters, Sigma-Delta modulation and many other. While the $task_i$ on signal is set to f , both components are disconnected to the state capacity. For digital cores the state value has to be converted once, while pulling, and back, while pushing. All the rest of the implementation remains the same.

Lastly, as for the Global Hormone Level signal line, a two-fold redundancy can be applied by doubling the state line and using two cross-connected decision modules.

3.2.3.3 Migrating between Domains

In [vRSH⁺15, Ch. 4.3] the essential process of transferring task between analog and digital cores has been described. Figure 3.12 shows the needed hardware for the analog and digital cores, while Figure 3.13 illustrates the concept of a mixed-signal task migration. Two analog and one digital core are used to distribute two tasks among them:

1. sonar sensor data read-out and filtering
2. power amplification for valve driving

An analog core, the analog controller, initially allocates the *sensor data read-out and filtering* task and connects to the sonar sensor. The *valve driving* task is allocated by the second analog core, a power amplifier. Caused by a monitored event, the first analog core drops its task, indicated with the red colored connection to the hormone bus. No other but the digital core, the digital controller, is available for reallocation. After reallocating, the data as well as the state, if available, are converted from analog to digital once and afterwards back again. However, it is to expect that state transfers may not be needed often [vRSH⁺15, Ch. 4.3].

Chapter 6.3 shows such a reallocation of a task between both domains. The analog PID controller is dropping its assigned task, due to a monitored failure. The task is picked up shortly afterwards by the digital PID controller.

4

Dependability Analysis

The open design measures, revealed in Chapter 3.2.2.4, identify the dependencies and ratios of the hormones G_i , $S_{\gamma,i}$, $A_{\gamma,i}$, E_γ and $\theta_{\gamma,i}$ as most important measures to derive, if AAHS is feasible or not. However, for a design analysis to state the feasibility the following set of constraints have to be considered also:

- Any task has to be allocated within its real-time bounds,
- No stable double allocations may occur,
- A task dropped by a core has to be reallocated by another,
- All constraints must occur for any number of cores (N) and any numbers of task $m < N$ at all times.

Next, a failure classification is needed to state the effects, which different failures have upon the task distribution and the whole system. The crucial failures lead to the dysfunction of the complete system and are to be eliminated, at least minimized. Others affect only the task distribution, while the already running system keeps operating. The least severe failure have no effect at all, since redundancy and the decentralized task distribution neutralized them.

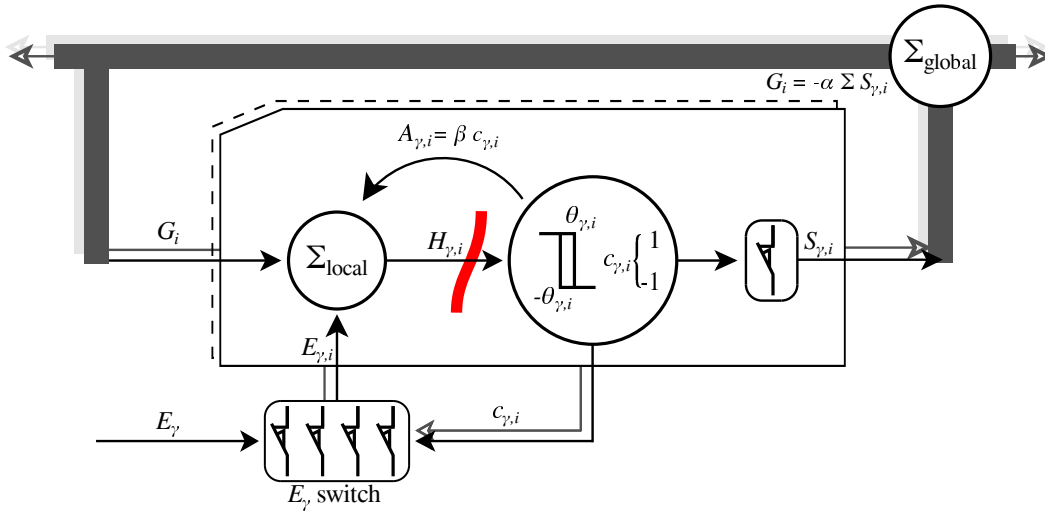


Figure 4.1: Sketch of the Analog Hormone System

4.1 Design Analysis

To determine the feasibility of the approach and extract any specifications needed for the design of the components, the analog hormone system is symbolically analyzed. Such an analysis states the interval of satisfying values of the hormones G_i , $S_{\gamma,i}$, $A_{\gamma,i}$, E_γ and $\theta_{\gamma,i}$ and much more interestingly the relations of the parameters on block level, a set of parameter dependencies. The analog hormone loop, as described in Chapter 3.2.1, is re-sketched in Figure 4.1 visualizing the symbolically mapped hormones and the different functions of the system:

- The Schmitt Trigger hysteresis $\theta_{\gamma,i} : \theta_{\gamma,i} \in \mathbb{R}^+$,
- The inner loop ($A_{\gamma,i} = \beta c_{\gamma,i} : \beta \in \mathbb{R}^+$) and its gain,
- The outer loop ($G_i = \sum_{\gamma=1}^N (-\alpha S_{\gamma,i}) : \alpha \in \mathbb{R}^+$) and its gain,
- The eager values $E_\gamma : E_\gamma \in \mathbb{R}^+$,
- The interval limits are defined as:

- $\alpha, \beta, E_\gamma, \theta_{\gamma,i} = (0, 1)$,
- $A_{\gamma,i}, H_{\gamma,i} = (-1, 1)$,
- $G_i = (-1, 0]$.

The parameter $c_{\gamma,i} = \{-1, 1\}$ defines the output of the Schmitt Trigger symbolizing the core taking a task or not ($C_{\gamma,i} = \{\mathbf{t}, \mathbf{f}\}$). Similar, the suppressor $S_{\gamma,i}$

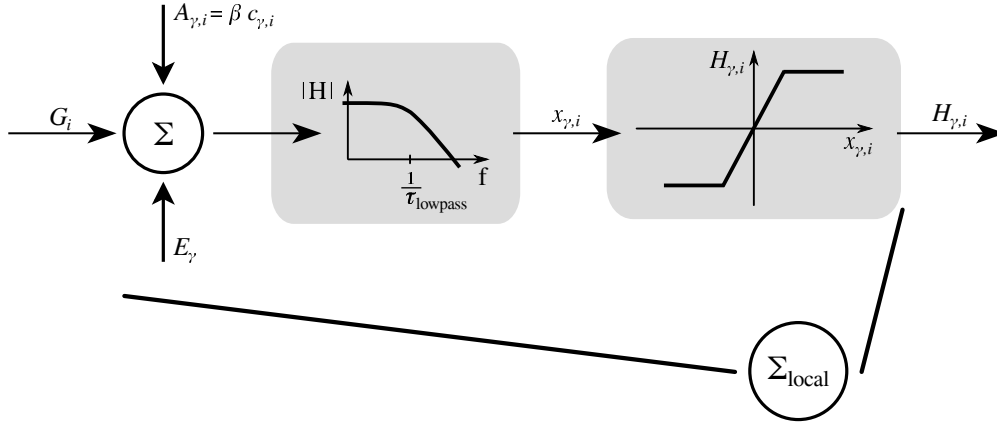


Figure 4.2: Block Diagram of a Limiting Adder Circuit

is defined by $\{1, 0\}$, either suppressing, while the core is turned on, or being idle. The parameter β defines the factor of the inner loop to locally keep a task alive, while globally the hormone level G_i drops. The drop (ΔG_i) calculates the new $\eta_i = \sum_{\gamma=1}^N -\alpha S_{\gamma,i} = -\alpha \sum_{\gamma=1}^N S_{\gamma,i}$, while α symbolized the ratio each suppressor hormone $S_{\gamma,i}$ influences the Global Hormone Level G_i compared to the Local Hormone Level $H_{\gamma,i}$ of every core.

The full function of the decision module can be described by differential equations. The first differential equation

$$x_{\gamma,i} + \tau_{\text{lowpass}} \cdot \dot{x}_{\gamma,i} = (\beta c_{\gamma,i} + G_i + E_\gamma) \quad (4.1)$$

$$H_{\gamma,i} = \begin{cases} 1 & \text{if } x_{\gamma,i} > 1, \\ -1 & \text{if } x_{\gamma,i} < -1, \\ x_{\gamma,i} & \text{otherwise} \end{cases}$$

describes the function of the limiting adder according to the block diagram of Figure 4.2, where $x_{\gamma,i}$ is an internal signal and τ_{lowpass} represents the time constant of the low-pass filter and equating the Local Hormone Level $H_{\gamma,i}$.

The next differential equation should describe the function of an inverting Schmitt Trigger. However, to simplify the total set of equations, a case distinction also models the output of the Schmitt Trigger

$$c_{\gamma,i} = \begin{cases} 1 & \text{if } H_{\gamma,i} < -\theta_{\gamma,i}, \\ -1 & \text{if } H_{\gamma,i} > \theta_{\gamma,i}, \\ c_{\gamma,i} & \text{otherwise.} \end{cases} \quad (4.2)$$

and reduces the complexity. The behavior of the Schmitt Trigger is abstracted, since the hormone loop will be untied at position of the Local Hormone Level $H_{\gamma,i}$. Equation (4.1) and (4.2) represent an AAHS decision module. The output of the adder affects the Schmitt Trigger, while the Schmitt Trigger also affects the adder function with ratio β instantly.

Next to the module, the Global Adder has to be defined. The differential equations of the adder function of the outer loop $\sum_{\gamma=1}^N -\alpha S_{\gamma,i} = -\alpha \sum_{\gamma=1}^N S_{\gamma,i}$ are

$$\begin{aligned}
 S_{\gamma,i} &= \begin{cases} 1 & \text{if } c_{\gamma,i} > 0, \\ 0 & \text{otherwise,} \end{cases} \\
 G_i + \tau_{G,i} \cdot \dot{G}_i &= -\alpha \sum_{\gamma=1}^N S_{\gamma,i} \text{ and} \\
 \eta_i &= \begin{cases} 0 & \text{if } G_i > 0, \\ -1 & \text{if } G_i < -1, \\ G_i & \text{otherwise,} \end{cases}
 \end{aligned} \tag{4.3}$$

which implements an adder of the switched suppressor hormones $-\alpha S_{\gamma,i}$ and a limiting to the signal range.

4.1.1 Algebraic Description

Those differential equations (4.1) to (4.3) must characterize every core within the hormone system for any time during runtime. This leads to a huge set of equations depending upon another and themselves over time. To simplify this, three assumptions have to be made:

1. The change of time within those equations is eliminated assuming that the differential equations settle to a limit value. The system is solved at specified times having these settlements in mind.
2. Also assuming that $\tau_{L,\gamma,i} < \tau_{G,i}$ applies, enabling the restriction of the time to specified points.
3. After the adder of the inner loop and the determination of the trigger (at position of the Local Hormone Level $H_{\gamma,i}$, see Figure 4.1), the system is untied to open the inner and outer loop.

The differential equations and the constraints generate a set of inequalities describing the AAHS. By solving those inequalities, the feasible region and the

relations of the system parameters $\theta_{\gamma,i}$ to α , β and E_γ , which applies for N cores and m tasks, are defined. To formalize the constraints between taking a task or not and keeping or loosing it afterwards, multiple time points t_k , abstracted to $t_k \in \mathbb{N}$, have to be considered.

Generally, a task T_i can never be assigned to two cores, but all cores eager to allocate task T_i must fulfill equation (4.4).

$$\theta_{\gamma,i} \leq \beta c_{\gamma,i}|_t + G_i|_t + E_\gamma|_t \quad (4.4)$$

Lets assume that core C_γ is the quickest and/or the most eager core to allocate task T_i , core C_γ fulfills equation (4.4) the first. The Global Hormone Level drop ΔG_i represents the value the Global Hormone Level is suppressed by at $t+1$. If $\Delta G_i \equiv \sum_{\gamma=1}^N -\alpha S_{\gamma,i} \equiv -\alpha S_{\gamma,i}$ the Global Hormone Level $\eta_i \equiv G_i|_{t+1}$ is suppressed by one core only, indicating that task T_i is allocated once. An implicit change within the inequalities is the case:

$$-\theta_{\gamma,i} \leq \beta c_{\gamma,i}|_{t+1} + G_i|_{t+1} + E_\gamma|_{t+1} \quad \text{and} \quad (4.5)$$

$$\forall_{\kappa \neq \gamma} : \beta c_{\kappa,i}|_{t+1} + G_i|_{t+1} + E_\kappa|_{t+1} < \theta_{\kappa,i}. \quad (4.6)$$

Equation (4.5) is valid for the core C_γ having taken task T_i at t . Any other core $C_{\kappa \neq \gamma}$ forsakes to assign that specific task at $t+1$ (equation (4.6)). If more than one core had simultaneously decided to allocate a task at any time point, (4.5) could not be fulfilled, due to the increased $\Delta G_i|_{t+1}$ drop as shown in equation (4.7). Those cores, which allocated task T_i would discard the task, and a new allocation process starts again. However, a reallocation of a task between two cores can occur to a later time, though a task can only be executed by one core at any time.

$$N > 2$$

$$\Rightarrow \Delta G_i|_{t+1} = \sum_{i=1}^N (-\alpha S_{\gamma,i}) \quad (4.7)$$

In an idealistic environment, the discard of task T_i by all cores would lead to an oscillation, but the time constant $\tau_{G,i}$ within the differential equations suffers marginal differences due to process variation, implementing an indirect priority list.

A small example of a task reallocation clarifies the behavior of equations (4.4) till (4.6). A reallocation caused by a core breakdown ($E_3|_{t=2} = 0$) of task T_i takes place, if

$$-\theta_{\gamma,i} \leq \beta c_{3,i}|_{t=2} + G_i|_{t=2} + 0 \quad (4.8)$$

Listing 4.1: Complete Algebraic Analysis.

```

1 define all basic constraints
2
3 for all time steps
4   determine all allocation constraints;
5   create the Inequalities;
6
7 solve for feasible sets;
8 calculate the radius  $r_{CC}$  and the coordinates of the Chebyshev center;
9 generate the specifications;

```

fails and core C_3 abandons task T_i . The new $\eta_i \equiv G_i|_{t=3}$ now causes that e.g. core C_1 fulfills (4.4) and allocates task T_i .

A Complete Algebraic Analysis The complete algebraic analysis, done with Maple [Map17], is described by the Pseudo-Code of Listing 4.1¹. The analysis starts with defining the basic constraints:

- Number of cores N and tasks m ,
- Defining $H_{\gamma,i} = -G_i + E_\gamma + A_{\gamma,i}$ for all cores N and tasks m ,
- $-1 < H_{\gamma,i} < 1$,
- $0 < \alpha, \beta, \theta, E < 1$,
- $3\alpha < 1$.

Having the free variables $\alpha, \beta, \theta, E_\gamma$ aligned between $(0, 1)$ eliminates any sign difficulty. The mathematical operations indicate the sign of the free variables and with the constraint $-1 < H_{\gamma,i} < 1$, it is stated that all operations are within the interval limits $(-1, 1)$. The constraint $3\alpha < 1$ implies that even if three cores are simultaneously allocating the same task, the triple suppressor value is not reaching the upper bound (saturation). Equation (4.7) demands a task drop, if a double allocation occurs. Hence, 3α indicates the suppression of a double allocation and a sufficient safety margin before reaching the upper bound. Due to the equation (4.7) triple and more allocations also lead to a task drop.

Next to the definition of the basic constraints, the huge set of equations and inequalities have to be defined. Equations (4.4) to (4.6) are repeatedly defined for all tasks m and cores N for different time steps. At least three time steps have to be considered:

¹A more details description is given in Appendix A.1

1. $t = 0$: All cores are turned off, no task is allocated, all cores are eager to allocate a task (equations (4.4)).
2. $t = 1$: For each task T_i only one core allocates that task, fulfilling equation (4.5), while equation (4.4) must be true for every other core $\kappa \neq \gamma$ the task T_i is not assigned to.
3. $t = 2$: To allow a task reallocation, a new task transfer is issued. All occupied cores drop their tasks by dissatisfying equation (4.5), while the others fulfill equation (4.4) again. A task allocation process as in step 2, follows.

Noticeable, $Hd_{\gamma,i}$ of Line 6 in Listing A.1, based on equation (4.7), is included at every time step to ensure a task discard, if more than one core allocates the same task. The gain of this addition is that, if a feasible region is found, the region automatically implies that the tasks are allocated only once. Once the huge set of equations and inequalities covering all cases regarding run-time, N cores and m tasks are defined, a simplex solver is used to calculate the feasible region of the AAHS. Lastly, the Chebyshev Center is calculated to determine the center of the maximum sized hypersphere, which fits inside the polyhedron without cutting the convex hull. With the determination of the values α , β , θ and r_{CC} the specifications of the components are extracted, as described in detail in Chapter 5.1.

The functions and their detailed description can be found in Appendix A.1.

4.1.2 Symbolic Solution

Each initial set of equations and inequalities is defined by the amount of active cores and tasks, for example $C_\gamma \in \{1..3\}$, $T_i \in \{1,2\}$. The columns of Table 4.1 represent the initial set differing only by the amount of active cores N , which want to allocate m tasks, which are to be allocated. Further, for each predefined value of the fixed variable (either $\theta_{\gamma,i}$ or E_γ) the huge set of equations and inequalities has to be evaluated. This leads to as many new, slightly differing sets as predefined values are defined. For each of this newly defined sets the Chebyshev Center algorithm calculates the coordinates of the Chebyshev Hypersphere, including the radius r_{CC} . Each row of Table 4.1 indicates the new set, differing by the predefined value of the fixed variable.

Analyzing Table 4.1 shows, as predicted and wanted, that the AAHS is scalable. For any $N > 2$ and $m < N$ the values of α , β , E_γ are only depending on $\theta_{\gamma,i}$. Further, the following statements can be made:

Table 4.1: Evaluation of the Symbolic Analysis

	$N = 2, m = 1$	$N = 4, m = 3$	$N = 5, m = 2$	$N = 7, m = 6$
$\theta_{\gamma,i} = 0.03$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$
	$\alpha = 0.269$	$\alpha = 0.269$	$\alpha = 0.269$	$\alpha = 0.269$
	$\beta = 0.1147$	$\beta = 0.1147$	$\beta = 0.1147$	$\beta = 0.1147$
	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$
$\theta_{\gamma,i} = 0.04$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$
	$\alpha = 0.269$	$\alpha = 0.269$	$\alpha = 0.269$	$\alpha = 0.269$
	$\beta = 0.1047$	$\beta = 0.1047$	$\beta = 0.1047$	$\beta = 0.1047$
	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$
$\theta_{\gamma,i} = 0.05$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$
	$\alpha = 0.269$	$\alpha = 0.269$	$\alpha = 0.269$	$\alpha = 0.269$
	$\beta = 0.0947$	$\beta = 0.0947$	$\beta = 0.0947$	$\beta = 0.0947$
	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$
$\theta_{\gamma,i} = 0.06$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$
	$\alpha = 0.269$	$\alpha = 0.269$	$\alpha = 0.269$	$\alpha = 0.269$
	$\beta = 0.0847$	$\beta = 0.0847$	$\beta = 0.0847$	$\beta = 0.0847$
	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$
$\theta_{\gamma,i} = 0.07$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$
	$\alpha = 0.269$	$\alpha = 0.269$	$\alpha = 0.269$	$\alpha = 0.269$
	$\beta = 0.0747$	$\beta = 0.0747$	$\beta = 0.0747$	$\beta = 0.0747$
	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$
$\theta_{\gamma,i} = 0.08$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$
	$\alpha = 0.269$	$\alpha = 0.269$	$\alpha = 0.269$	$\alpha = 0.269$
	$\beta = 0.0647$	$\beta = 0.0647$	$\beta = 0.0647$	$\beta = 0.0647$
	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$	$E_{\gamma} = 0.2357$
$\theta_{\gamma,i} = 0.09$	$r_{CC} = 0.0618$	$r_{CC} = 0.0618$	$r_{CC} = 0.0618$	$r_{CC} = 0.0618$
	$\alpha = 0.2714$	$\alpha = 0.2714$	$\alpha = 0.2714$	$\alpha = 0.2714$
	$\beta = 0.0618$	$\beta = 0.0618$	$\beta = 0.0618$	$\beta = 0.0618$
	$E_{\gamma} = 0.2394$	$E_{\gamma} = 0.2394$	$E_{\gamma} = 0.2394$	$E_{\gamma} = 0.2394$
$\theta_{\gamma,i} = 0.10$	$r_{CC} = 0.059$	$r_{CC} = 0.059$	$r_{CC} = 0.059$	$r_{CC} = 0.059$
	$\alpha = 0.2739$	$\alpha = 0.2739$	$\alpha = 0.2739$	$\alpha = 0.2739$
	$\beta = 0.059$	$\beta = 0.059$	$\beta = 0.059$	$\beta = 0.059$
	$E_{\gamma} = 0.2432$	$E_{\gamma} = 0.2432$	$E_{\gamma} = 0.2432$	$E_{\gamma} = 0.2432$
$\theta_{\gamma,i} = 0.11$	$r_{CC} = 0.0568$	$r_{CC} = 0.0568$	$r_{CC} = 0.0568$	$r_{CC} = 0.0568$
	$\alpha = 0.2765$	$\alpha = 0.2765$	$\alpha = 0.2765$	$\alpha = 0.2765$
	$\beta = 0.0568$	$\beta = 0.0568$	$\beta = 0.0568$	$\beta = 0.0568$
	$E_{\gamma} = 0.2471$	$E_{\gamma} = 0.2471$	$E_{\gamma} = 0.2471$	$E_{\gamma} = 0.2471$

- Beginning with a low threshold voltage $\theta_{\gamma,i} = 0.03$ the radius r_{CC} of the Chebyshev Hypersphere is calculated to be $r_{CC} = 0.0643$. The threshold difference of $[\theta_{\gamma,i}, -\theta_{\gamma,i}]$ equals to 0.06, a little less than r_{CC} . This implies that $\theta_{\gamma,i} < 0.03$ should be avoided.
- As $\theta_{\gamma,i}$ increases, the value of β decreases, indicating the close relation between $\theta_{\gamma,i}$ and β .
- The maximum of all three variables $\theta_{\gamma,i}, \beta, r_{CC}$ is reached at $\theta_{\gamma,i} = 0.07$ with $\beta = 0.0747$ and $r_{CC} = 0.0643$. It is to be assumed that this is the optimal assignment for a fixed $\theta_{\gamma,i}$ and variable $\alpha, \beta, E_{\gamma}$.
- With an increasing $\theta_{\gamma,i} \geq 0.09$ the values of β and r_{CC} are equal. Now, the decreasing β is the hardest constraint of the Chebyshev Center. Any further increase by $\Delta\theta_{\gamma,i} = 0.01$ indicates a $\Delta r_{CC} = 0.003$ decrease.

Next, evaluations have been done with the combined sets of differing $\theta_{\gamma,i}$ values. The results are presented at Table 4.2 and state that the hard constraints of the respective sets apply together at the combined sets, even worsening the results. Table 4.3 further confirms this assumption. Any $\theta_{\gamma,i} \geq 0.09$ implies that $\beta = r_{CC}$.

As already assumed the evaluation done for $\theta_{\gamma,i} = 0.07$ provides the optimum in regard of $\theta_{\gamma,i}, \beta$ and r_{CC} . Though, recalling the Figures 3.7, 3.8 on pages 65, 67 the eager values are provided and adjusted by external voltage sources, which are commonly adjustable in 50 mV steps. Equation (4.9) calculates the actual needed voltage for the eager value according to the mean of the supply voltage of the used technology².

$$E_{\gamma}(V) = \frac{V_{DD}}{2} \cdot 0.2357 \quad (4.9)$$

$$E_{\gamma}(V) = 0.3889 V$$

$$\therefore [0.3889 V] = 0.4 V$$

$$\Rightarrow E_{\gamma} = 0.2424 \quad (4.10)$$

With the new value of E_{γ} (equation(4.10)) a complete re-run of the symbolic analysis is issued, with E_{γ} being fixed at 0.2424 and α, β and $\theta_{\gamma,i}$ are free variables. Table 4.4 shows the change of $\theta_{\gamma,i}, \alpha, \beta$ and r_{CC} in regard to the fixed $E_{\gamma} = 0.2424$, compared to the chosen optimum.

²See Chapter 5.1 for more details on the used design technology.

Table 4.2: Evaluation of Combined Sets

	$N = 2, m = 1$	$N = 4, m = 2$	$N = 7, m = 6$
$\theta_{\gamma,i} = [0.06, 0.07]$	$r_{CC} = 0.0623$	$r_{CC} = 0.0623$	$r_{CC} = 0.0623$
	$\alpha = 0.2709$	$\alpha = 0.2709$	$\alpha = 0.2709$
	$\beta = 0.0803$	$\beta = 0.0803$	$\beta = 0.0803$
	$E_{\gamma} = 0.2386$	$E_{\gamma} = 0.2386$	$E_{\gamma} = 0.2386$
$\theta_{\gamma,i} = [0.07, 0.08]$	$r_{CC} = 0.0623$	$r_{CC} = 0.0623$	$r_{CC} = 0.0623$
	$\alpha = 0.2709$	$\alpha = 0.2709$	$\alpha = 0.2709$
	$\beta = 0.0703$	$\beta = 0.0703$	$\beta = 0.0703$
	$E_{\gamma} = 0.2386$	$E_{\gamma} = 0.2386$	$E_{\gamma} = 0.2386$
$\theta_{\gamma,i} = [0.08, 0.09]$	$r_{CC} = 0.0618$	$r_{CC} = 0.0618$	$r_{CC} = 0.0618$
	$\alpha = 0.2714$	$\alpha = 0.2714$	$\alpha = 0.2714$
	$\beta = 0.0618$	$\beta = 0.0618$	$\beta = 0.0618$
	$E_{\gamma} = 0.2394$	$E_{\gamma} = 0.2394$	$E_{\gamma} = 0.2394$
$\theta_{\gamma,i} = [0.09, 0.1]$	$r_{CC} = 0.059$	$r_{CC} = 0.059$	$r_{CC} = 0.059$
	$\alpha = 0.2739$	$\alpha = 0.2739$	$\alpha = 0.2739$
	$\beta = 0.059$	$\beta = 0.059$	$\beta = 0.059$
	$E_{\gamma} = 0.2432$	$E_{\gamma} = 0.2432$	$E_{\gamma} = 0.2432$

Table 4.3: Evaluation of the Set $\theta_{\gamma,i} = [0.05, 0.08]$

	$N = 2, m = 1$	$N = 5, m = 2$	$N = 6, m = 4$
$\theta_{\gamma,i} = [0.05, 0.08]$	$r_{CC} = 0.0583$	$r_{CC} = 0.0583$	$r_{CC} = 0.0583$
	$\alpha = 0.2747$	$\alpha = 0.2747$	$\alpha = 0.2747$
	$\beta = 0.0817$	$\beta = 0.0817$	$\beta = 0.0817$
	$E_{\gamma} = 0.2444$	$E_{\gamma} = 0.2444$	$E_{\gamma} = 0.2444$

Table 4.4: Evaluation Comparison of $\theta_{\gamma,i}$ and E_{γ}

Fixed variable	$\theta_{\gamma,i} = 0.07$	$E_{\gamma} = 0.2424$
Free variables	$r_{CC} = 0.0643$	$r_{CC} = 0.0611$
	$\alpha = 0.269$	$\alpha = 0.272$
	$\beta = 0.0747$	$\beta = 0.0611$
	$E_{\gamma} = 0.2357$	$\theta_{\gamma,i} = 0.0911$

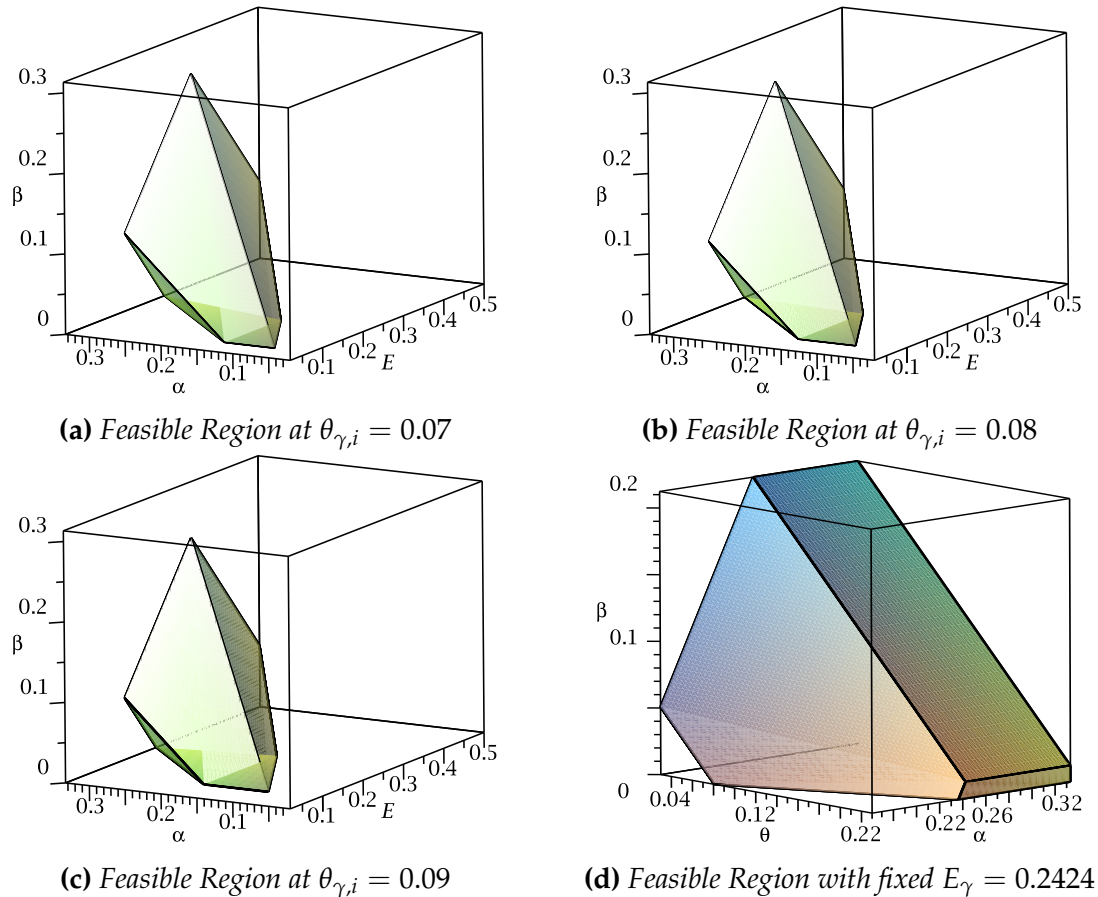


Figure 4.3: Four Polyhedrons Representing Different Feasible Regions

Figure 4.3 shows the feasible region of three different $\theta_{\gamma,i}$, while the fourth shows the region with the fixed $E_{\gamma} = 0.2424$. The differences of the three solved with a fixed θ are marginal, the coordinates of the Chebyshev Center vary only in β . Substituting E_{γ} for $\theta_{\gamma,i}$ as fixed variable reveals a new perspective, as Figure 4.3(d) shows. Building on the polyhedron with fixed $E_{\gamma} = 0.2424$, the hypersphere with the Chebyshev Center is shown in Figure 4.4, touching the convex hull, but never exceeds.

At last, a corner case analysis for $N = m$ is given in Table 4.5. The reason for consideration is given by the evaluation of the design validation in Chapter 6.1. The resulting values for different $\theta_{\gamma,i}$ of the Chebyshev Center calculation have not changed, even though the polyhedrons differ, as Figure 4.5 shows. However, the similarity of the Chebyshev Center was exceeded due to two cases:

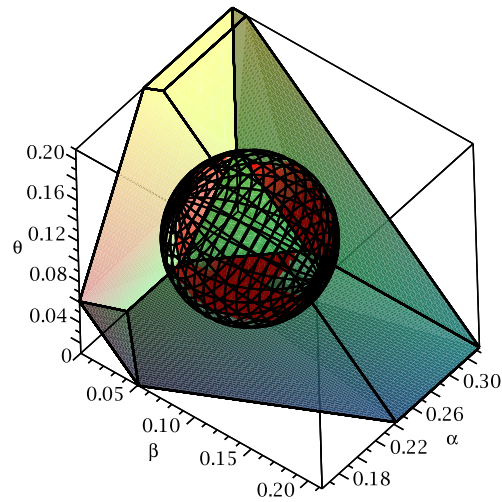


Figure 4.4: The Chebyshev Sphere inside the Polyhedron with Fixed $E_\gamma = 0.2424$

Table 4.5: Evaluation of the Corner Case Analysis

	$N = 2, m = 2$	$N = 3, m = 3$	$N = 5, m = 5$	$N = 7, m = 7$
$\theta_{\gamma,i} = 0.03$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$
	$\alpha = 0.269$	$\alpha = 0.269$	$\alpha = 0.269$	$\alpha = 0.269$
	$\beta = 0.1147$	$\beta = 0.1147$	$\beta = 0.1147$	$\beta = 0.1147$
	$E_\gamma = 0.2357$	$E_\gamma = 0.2357$	$E_\gamma = 0.2357$	$E_\gamma = 0.2357$
\vdots	\vdots	\vdots	\vdots	
$\theta_{\gamma,i} = 0.08$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$	$r_{CC} = 0.0643$
	$\alpha = 0.269$	$\alpha = 0.269$	$\alpha = 0.269$	$\alpha = 0.269$
	$\beta = 0.0647$	$\beta = 0.0647$	$\beta = 0.0647$	$\beta = 0.0647$
	$E_\gamma = 0.2357$	$E_\gamma = 0.2357$	$E_\gamma = 0.2357$	$E_\gamma = 0.2357$

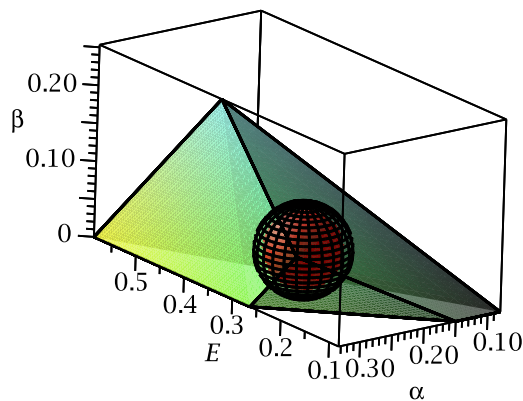


Figure 4.5: Polyhedron representing Feasible Region of the Corner Case $m = N$

1. The set of inequalities misses out on only one constraint, a cutting facet, due to the forcing symmetrical distribution of all tasks to all cores, leaving no free core available.
2. Limited by the case of untying the hormone loop before the Schmitt Trigger (Figure 4.1) the allocation process itself is neglected. The analysis regards the state before the allocation and afterwards, but the changes of the inequalities during the process are ignored.

4.2 Reliability Analysis

To be able to state the reliability gain of the analog hormone system, the failure susceptibility of the system has to be analyzed. The Failure Severity Diagram of Figure 1.2 on page 5 allows to classify, which failure modes AAHS is able to handle, if the system itself is reliable. However, a small degradation scenario with three cases states the need to analyze the failure susceptibility of the architecture as well:

- The system degrades uniformly, equally affecting all components. The ratio of the parameters α , β , E and θ remain close to constant, implying that the system operates within the feasible region [vRSH⁺15].
- The degradation affects only partially one core due to e. g. heat, which has a task allocated. A monitor detects the heat, decreases the eager value and initiates the task reallocation [vRSH⁺15].
- The most interesting case is, if the degradation applies uneven to the whole system. The ratios of α , β , E and θ are beginning to vary at each core, changing the allocation process unpredictably. As long as the variation stays within the feasible region, the distribution system stays operational, maybe not within the optimum. However, the worst case would be, if the degradation leads to the total loss of the distribution system [vRSH⁺15].

Only the first case is good-natured, the others demand monitor circuits within the hormone system as well. For example, AAHS relies heavily on the Schmitt Triggers and their threshold voltages. Figure 4.6 shows the change of the trigger behavior over time caused by constant stress, compared to a fresh trigger and one recovering. As the trigger of core γ degrades, the hysteresis worsens, θ_γ increases. The ratio of θ to the other three variables α , β , E clearly worsened. However,

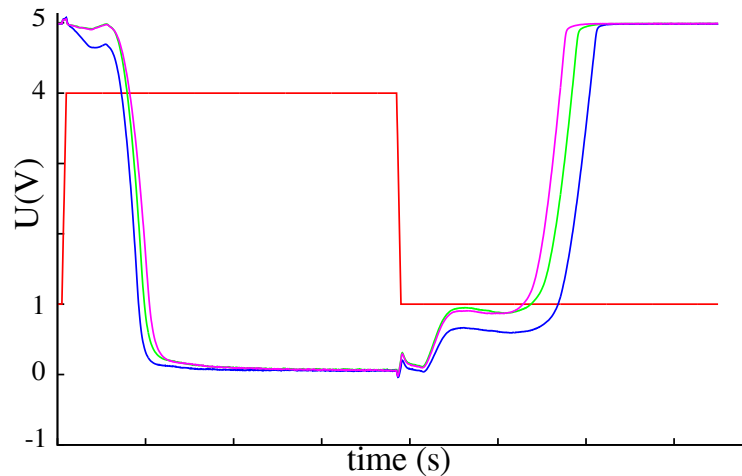


Figure 4.6: *Degradation Effects of an Inverted Schmitt Trigger [vRSH⁺15] with the Pulse Signal (red), Fresh Signal (green) and the Aging Signal (blue), and the free-of-aging Signal (violet)*

not only does the Schmitt Trigger directly influence the distribution system, the other components need to be investigated as well (Chapter 4.2.2). The reliability analysis allows to classify the components according to their fail-safety, defining failure impact classes.

4.2.1 Failure Classification

Based on [Shu13], a classification model to state reliability and safety of the AAHS approach is developed. Three classes are defined:

Failure Class A: Failing effects, which lead to an immediate fail of the distribution system and the full system.

Failure Class B: Failing effects, which lead to the loss of the reliability, since the hormone loop is ignored. However, the task is being executed at the core last allocated and will remain active until the core dies (if at all).

Failure Class C: Failing effects, which influence the local decision module of the AAHS only. The hormone loop stays active, just ignoring the one module. The reliability is still ensured, the distribution process issued and the full system operating.

In Table 4.6 the failure class of each component $\chi \in \{1..12\}$, as enumerated in Chapter 4.2.2 and Figure 4.7, is named. A distinction is done for some components based on their different failing cases (a,b,c - see pages 96, 97).

Table 4.6: *Failure Class Occurrences*

Failure Class	Classification of each component χ
A	7, 8, 11b, 14
B	1b, 2b, 3b, 4a, 4b, 10a, 10b
C	1a, 2a, 3a, 4c, 5, 6, 9, 10c, 11a, 12, 13

The failing effect of Failure Class C can be neglected. The architecture itself absorbs the failures. Without monitors these failing effects pass unnoticed, which is however dangerously in certain circumstances. Any failing decision module of the AAHS must be replaced to hold the failing core tolerance F_t .

The impact of failures of Failure Class B can be weakened by optimizing for example the lifetime of the cores. However, idle, useless and redundant cores and a dysfunctional distribution system are an irredeemable size overhead. Increasing the amount of decision modules does not catch such failures either. Monitoring the circuits and detecting these failing effects beforehand, allows to proactively migrate the task to an healthy core. The failing effects can be downgraded to Class C effects with a functional AAHS architecture.

Any Class A failing effect has severe impacts on the full system and the AAHS, and needs to be regarded as a single point of failure. Monitors detecting the failed components are useless, the system is already down. Monitoring the failing components increases the lifetime of the components for example through recovery phases, but only shortly. The affected components need to be replaced, but the replacement can not be done on-the-fly. For example the Global Adder unit can only be replaced, if the full system is powered down. More so, the components need to be designed reliably. However, the two-fold approaches, described in Chapter 3.2.2.2, eliminate the global units as single points of failure, which leads to a change of Table 4.6, as equation (4.11) and (4.12) show. Only the supply network remains as single point of failure.

$$\text{Failure Class } A = A \setminus \{7, 8, 11b\}, \quad (4.11)$$

$$C = C \cup \{7, 8, 11b\}. \quad (4.12)$$

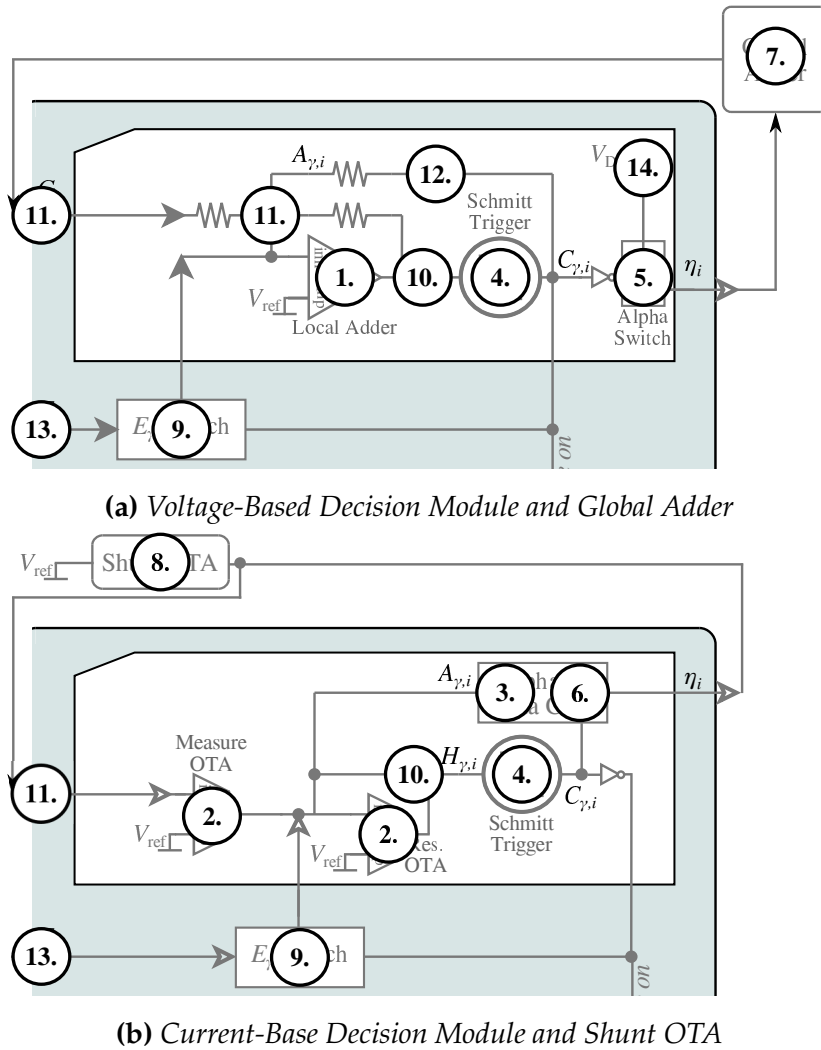


Figure 4.7: Classifying the Fail-Safety of the Decision Modules

However, for the scope of this thesis the components 7, 8 and 11b remain as single points of failure.

4.2.2 Fail-Safety Investigation

Investigating the fail-safety of AAHS implies the failure analysis of each component and the wiring network of the architecture. It allows to identify the critical areas/components of the approach and assign the failure classes. In [Shu13] a first investigation for AAHS has been done. Figure 4.7 shows both AAHS implementations with all components enumerated for the fail-safety investigation. To

be able to judge the fail-safety of a component, the following bullets are examined:

- Failure-aiding behavior,
- Influence of other components,
- Possible monitoring circuits,
- Effect on the AAHS, if failing.
- Effect on the full system, if failing.

Especially stating the difference failures have on the AAHS and the full system is of importance: Does a failure lead to the loss of either a decision module, the hormone loop of a task (affecting only the cores, which applied for that task), the complete distribution of all tasks or the full system? The loss of a task distribution loop or the complete AAHS does not automatically imply a complete system loss. It implies the loss of the reliability, but the tasks remain active on their cores for as long as the cores do not fail. The enumerations are defined as follows:

1. **Local Adder:** Failing Local Adders falsify the equation (4.4) till (4.6), immediately influencing the Local Hormone Level $H_{\gamma,i}$, leading to rejecting any tasks or never dropping an allocated task.
 - (a) Rejecting implies that the task T_i is allocated to other cores, the hormone loop and the full system stay operational [Shu13].
 - (b) Contrary, never dropping an allocated task T_i indicates the loss of reliability, the task is executed at the core until the core fails [Shu13].

Checking the difference of the input signals enables the monitor to decide between the two failure reactions. Most important, the negative difference hardens the dropping equation (4.5), since $H_{\gamma,i}$ increases. A monitor checking on the negative difference decreases E_{γ} respectively and counters the failing effect to some degree.

2. **Local OTAs:** The two local OTAs, Measure OTA and Res. OTA, can be classified in union in regard to fail-safety. Equivalent to the Local Adder failing performances of the OTAs lead to falsification of the equation (4.4) till (4.6), which influence $H_{\gamma,i}$ immediately. Two distinctions in terms of fail-safety have to be made:

- (a) The Local Hormone Level is not exceeding the threshold voltage of the ST anymore, the task T_i is allocated by other cores. The full system operation and the hormone loop is not confined.
- (b) A task drop can not be issued anymore, the reliability gain of the AAHS is lost. The task remains allocated by the core, stating Failure Class B .

Monitoring the local OTAs should focus on the performances which increase $H_{\gamma,i}$ primarily to avoid 2b. Decreasing E_γ respectively to the faulty increase allows to counter the failing effect, but only to some degree. Beyond that the reliability of AAHS is lost.

3. **Beta CM:** The Beta Current Mirrors feed the local loops with $\beta(A)$.

- (a) A loss of the Beta CM always forces a decision unit to lose an allocated task T_i , a reallocation process is issued. The hormone loop of the task T_i and the system stay operational.
- (b) Contrary, if a task drop can never be issued again, since β increases, the task T_i is executed at the core until the core dies. The hormone loop for this task is disabled.

However, a monitor checking on the value of the output β can disable the eagerness, if β increased beyond its feasible interval. As long as $\beta < E$ applies, setting $E = 0$ forces a task drop, the hormone loop stays operational (Failure Class C), but $\beta > E$ implies Failure Class B .

4. **Schmitt Trigger:** The Schmitt Triggers experience among other effects threshold voltage drifts and offset shifts due to NBTI and HCI stress caused by the varying Local Hormone Level $H_{\gamma,i}$. These changes worsen the performances of the Schmitt Triggers and may lead to faulty task allocation or drop behaviors. The failing effects are classified in three cases:

- (a) The Schmitt Trigger does not drop the task T_i , even though $H_{\gamma,i} < -\theta$ applies. The task distribution fails, but the full system stays operational [Shu13].
- (b) The Schmitt Trigger allocates the task T_i , even though $H_{\gamma,i} < \theta$ applies. The violation of the allocation constrains imply a loss of the functionality of the AAHS. Under worst case conditions the task T_i has been allocated before, the double allocation causes a task drop at both cores, initiating the allocation process with the failing Schmitt Trigger trig-

gering first. The task is allocated and the reliability lost, but the full system is operating.

- (c) The Schmitt Trigger does not trigger anymore, even though $\theta < H_{\gamma,i}$ applies. Such a decision module participates on the decision process, but never allocates a task. Neither the hormone loop is affected nor is the full system functionality harmed [Shu13].

Monitoring the Schmitt Trigger should lead to the detection of the violation of the constraints of equation (4.4) till (4.6). Further, the monitor circuit should initialize countermeasures to fulfill the constraints again.

5. **Alpha Switch:** The oversized switching transistor is operated rail-to-rail as a digital output signal, either on or off. It follows that the failing effects are:

- (a) An open implies that independent from the decision unit the suppressor will be zero, allowing unchallenged double allocations to occur. The AAHS and the full system is not operable anymore.
- (b) A short implies that a suppressor is constantly sent, the tasks dropped and never taken again. The hormone loop and the full system are considered dysfunctional.

The single transistor can hardly be monitored, but designed highly reliable. However, if a TDDDB effect can be detected by a monitor, the eager value E_{γ} of the decision module can be set to zero, allowing a Failure Class C classification.

6. **Alpha CM:** The Alpha Current Mirror pushes the suppressor current onto the hormone bus. Variation of the current value of α is acceptable, as long as the α interval is satisfied. However, if the interval is violated, especially if no suppressor is sent, although a suppressor should be sent, the hormone loop and the full system should not be operated anymore. Designing the current mirror highly reliable is most likely the best solution to keep the performances of the current mirror as proposed. Similar to the Alpha Switch, if the α interval is exceeded and detected by the monitor, the eager value E_{γ} can be set to zero to prevent a task allocation, classifying the mirror in Failure Class C.

7. **Global Adder:** Equivalent to the Local Adders, false output values falsify the equation (4.4) till (4.6). Though, either all trigger functions $\theta < H_{\gamma,i}$ are

allocating the task T_i or the task T_i is dropped and never again allocated. Either way, the hormone loop for this task and the full system are lost [Shu13]. Monitoring the offset drift of the input signals indicates the usability of the adder. If the offset exceeds the feasible interval of α , the adder must be replaced, which can not be done on-the-fly. In Chapter 3.2.2.2 a two-fold redundancy approach is presented to eliminate the loss of the hormone loop and the full system and enable the on-the-fly replacement of the adder.

8. **Shunt OTA:** A change of the potential difference between the negative input and the output of the OTA leads to a change of $H_{\gamma,i}$ at each core, falsifying the equation (4.4) till (4.6) at some point. Task T_i is either allocated or is discarded by all cores. Equivalent to the Global Adder, the distribution as well as the functionality of the full system is lost, if the Shunt OTA fails. Any monitor stating the degrading effect of the output signal (violating the feasible interval of α) notifies the need to replace the OTA and implies the need to transfer to the two-fold Shunt OTA (see Chapter 3.2.2.2).
9. **E_γ Switch:** The switching circuit is a simple exclusive or circuitry. The gates of the switching transistors are controlled by the Schmitt Trigger, operated rail-to-rail either opened or closed. Following failing effects could apply:
 - (a) Opens imply that a core is able to allocate different tasks simultaneously. The several hormone loops are affected by the faulty allocation and the full system is not operable anymore.
 - (b) Shorts imply that no eager values pass through, the involved decision modules are not participating on the allocation process anymore. The AAHS and the full system stay operational und fully functional.

Monitoring the individual transistors is hardly realizable, but the rail-to-rail operation enables a substantial reliability increase achieved by overdesigning. Severe failing effects, like TDDB, can be detected more easily, decreasing the eager value E_γ of the decision unit, allowing to classify the fail-safety as Failure Class C.

10. **Network N1:** The degradation of network N1 feeds the Schmitt Triggers with false values. The failure classification applies similar to the three cases of the Schmitt Trigger.
 - (a) The false values imply to hold on task T_i . The hormone loop fails, but the full system stays operating.

- (b) The false values lead to violating the task allocation constraints, similar to the Schmitt Trigger.
- (c) The false values hold off the Schmitt Trigger, which does not allocate task T_i anymore. The effect is locally, not affecting the hormone loop and the full system.

Monitoring the network should lead to a decrease of E_γ , lowering the participation of the decision module, classifying the failing network as Class C. Only the violation of the task allocation constraint remains classified as Failure Class B.

11. **Network N2:** Degrading effects on any wire of the network N2 influence the global suppressor value. As long as the feasible interval of α is not violated, the hormone system stays operational, but a violation implies the loss of reliability and the full system functionality. A network distinction for further consideration has to be made:
 - (a) The local network within the module,
 - (b) The global network corresponds to the hormone bus.

Monitoring the failure of the local wiring, for example an EM effect, allows to decrease the eager value E_γ , acknowledging a classification of the local network as Failure Class C. The two-fold redundancy of the global unit doubles the wires also. A monitor, for example a heart beat monitor, guards the wires and issues a switch between the global units until the faulty bus structure is replaced, reducing the failure class to Failure Class C, as it applies for the Global Adder (7) and Shunt OTA (8).

12. **Network N3:** Degrading of the local loop network behaves similar to the Beta CM failing effects. The loss of the network implies a loss of β , forcing to discard the allocated task T_i . However, the hormone loop and the full system stay operational, Failure Class C.
13. **Network N4:** Degrading wires of network N4 imply that the eager value E_γ is decreasing. As a result, the affected core will abandon any task it took and will not allocate any new tasks, and is classified as Class C.
14. **Supply Network:** The degradation or loss of the supply voltages V_{DD} , V_{ref} or the bias current I_{bias} indicate a complete loss of the full system, implying Failure Class A.

4.2.3 Failure Sensitivity

According to the Failure Class Model on page 92 and the failure modes described in Chapter 1.2 a sensitivity diagram for AAHS can be defined. This diagram marks, which components of the hormone system architecture need to be monitored for which failure mode to keep its functionality. Table 4.7 represents the diagram. The components of Failure Class *A* are the most critical components, any severe failures indicate the loss of the full system. The impact of the two-fold approach of the global hormone bus presented in Chapter 3.2.2.2 is of interest. The Global Adder and the Shunt OTA are modeled as working cores. As working cores, they are classifiable as redundant component. Further, the hormone line is double, forming a hormone bus. Next, the Alpha Switch and the Alpha CM can be downgraded in their failure class, if the monitor checks on the rightness of the output. Setting $E_\gamma = 0$ initiated by the monitor emits the decision module from the allocation process, similar to Failure Class *C*. Remaining as single point of failure is the supply voltage network, which needs to be closely monitored. However, any failing leads to a shutdown of the full system.

Further, the Schmitt Trigger is of interest, more precisely the scenario of the Schmitt Trigger, which violates the $H_{\gamma,i} < \theta$, respectively the $-\theta < H_{\gamma,i}$ constraints (Equations (4.5) and (4.6)). A Schmitt Trigger triggering to allocate task T_i while $V_{\text{threshold}} < \theta$, can not reliably be turned off with an decreasing eager E_γ . If task T_i has not been allocated before, this allocation process does no harm to the full system, but the reliability is lost. If task T_i is allocated already, a double allocation occurs now. The second suppressor leads to the drop of task T_i at both cores, issuing a new allocation process. Since $V_{\text{threshold}} < \theta$ applies at the failing Schmitt Trigger, it allocates first during the new process, beginning to execute task T_i . Again, reliability is lost, but the full system stays operational. Similar, the monitor initiates a task drop by setting $E_\gamma = 0$, yet no drop is issued. Task T_i is still executed at the core, but the reliability is lost. The other components of Class *B* can be downgraded to Failure Class *C*, if they are monitored and the eager value E_γ is decreased accordingly.

As already mentioned, AAHS absorbs any failing effect affecting a component initially classified in Failure Class *C*. Though, without any monitor failures pass unnoticed and dysfunctional decision modules are unreported, and will therefore not be replaced. Further, the robustness value r_{CC} defines the allowed variation factor of any component, neglecting process variation and the like. Figure 4.8

Table 4.7: Failure Susceptibility

Failure Classes A B C	Change in Failure Class	Need to monitor for			
		TE	NRD	EE	RD
Global Adder (7)		see Table 4.8			
Shunt OTA (8)		see Table 4.8			
Alpha Switch (5)	$A \rightarrow C$	Any process variation less than r_{CC} can be ignored, $r_{CC} =$ robustness value	yes	yes	no
Alpha CM (6)	$A \rightarrow C$		yes	yes	no
E_γ Switch (9)	$A \rightarrow C$		yes	yes	no
Network N2 (11a)	$A \rightarrow C$		yes	yes	no
Network N2 (11b)	Hormone Bus		yes	yes	no
Supply network (14)	-		no	yes	yes
Local Adder (1b)	$B \rightarrow C$		yes	yes	yes
Local OTAs (2b)	$B \rightarrow C$		yes	yes	yes
Beta CM (3b)	$B \rightarrow C$		yes	yes	yes
Schmitt Trigger (4a & 4b)	-		no	no	no
Network N1 (10a & 10b)	-	no	no	no	
Failure Class C components	No need to monitor, since negligible as AAHS itself absorbs any failure				

TE: Technology Effects, NRD: Non-Recoverable Degradation, EE: Environmental Effects, RD: Recoverable Degradation,

shows the Severity Diagram extracted to illustrate the failure sensitivity of the AAHS.

Next to the reliability analysis of AAHS, Table 4.8 shows the failure modes of the several sample cores, which AAHS is able to handle. Any component on this table can be multiplied and used as a core of the architecture.

The Global Adder and the Shunt OTA are added to the list of cores due to the two-fold approach (Chapter 3.2.2.2). The technology effects are handled by the robustness value r_{CC} of the hormone system. Monitoring the effects of degradation, either non-recoverable, environmental or recoverable, increases the reliability of AAHS, since for example depleted adder cores are marked as replaceable, while the second adder core keeps the system running. Sensitive to the failing effects are the offset and gain respectively G_m of the amplifiers, but also heat and radiation should be checked.

The other defined cores vary in the needed monitors, depending on their uses. For sure, additional cores can be designed and analyzed to state the failure han-

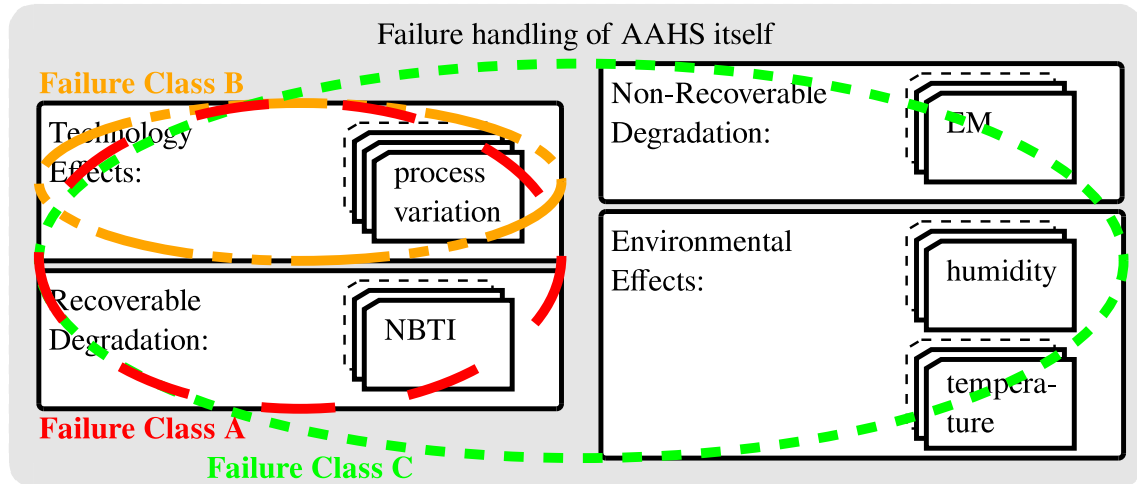


Figure 4.8: Sensitivity Diagram of the Analog Hormone System

dling of AAHS. However, the basic principle of the monitor circuits and the cores are that the failure handling is done by changing the eager value. Setting $E_\gamma = 0$ in case of a failure indicates the failed core. The dropped task will be taken by another core, ensuring dependability. Identifying the sensitive properties states an indicator the monitor should be checking for. For example the PID Controller monitors only the non-recoverable effects, since the loop to the ,actual state input' and the PID covers the other failure modes.

A couple of cores can increase their reliability independently by incorporating the redundancy with their design. Though, this does not influence the reliability of the overall architecture. Due to robustness value r_{CC} AAHS by itself is able to handle any failure, degradation and effect, where the variation change is less than r_{CC} . Design failures need to be checked during the design process (DRC, LVS, etc.).

In Chapter 6.2 follows a percentage area calculation based on equation (4.14) of the three failure classes comparing the area of the different classified components to the total area. $\square_{\text{Failure Class } \Psi}$ represents the sum of the area of the components χ classified as being in Failure Class $\Psi \in \{A, B, C\}$. This allows to state the probability of failure class occurrence, caused by randomly distributed failing effects and more interestingly, how likely Failure Class A will arise and lead to a complete system outage.

$$\square_{\text{Failure Class } \Psi} = \sum \square_\chi : \chi \in \Psi \quad (4.13)$$

$$P(X = \Psi) = \frac{\square_{\text{Failure Class } \Psi}}{\square_{\text{AAHS}}} : \Psi \in \{A, B, C\} \quad (4.14)$$

Table 4.8: Failure Sensitivity of Sample Working Cores

Working cores	Need to monitor for				Ability to incorporate redundancy	Properties sensitive to failures
	TE	NRD	EE	RD		
Global Adder	no, handled by r_{CC}	yes	yes	yes	no	Offset, Gain
Shunt OTA	no, handled by r_{CC}	yes	yes	yes	no	Offset, G_m
Low-pass Filter	yes	yes	yes	yes	yes	Offset
PID Controller	yes	yes	no	no	no	-
ADC/DAC	yes	yes	yes	yes	yes	Differential & integral nonlinearity
Output stage	yes	yes	yes	yes	yes	Offset, Gain
Monitor Core	yes	yes	yes	yes	no	Offset
Supply voltage battery core	yes	yes	yes	yes	yes	Voltage supply

TE: Technology Effects, NRD: Non-Recoverable Degradation,
 EE: Environmental Effects, RD: Recoverable Degradation,

4.2.4 Monitor Decisions

With monitor circuits controlling and adjusting the eager values E_γ of the cores, self-reliance of a system is attainable. However, with monitors deciding on E_γ of the cores autonomously, faulty decisions can occur, which are classified by the following four categories:

True positive: Any failure occurred at a core and the monitors reacted to turn down E_γ . In any case the core C_γ abandons its tasks, if it had some. This is how it is supposed to be.

False positive: No failure occurred, but the monitor of core C_γ triggered erroneously. The eager value E_γ is decreased and C_γ abandons any task, if allocated. This state is resentful, but manageable and AAHS stays operational.

False negative: This is the worst case category, since failures occurred, but no monitor triggers. The task execution is faulty, but unchallenged. Therefore, this category should be safely guarded and counteracted under all circumstances.

True negative: No failure occurred so far and no monitor reacted, as it is supposed to be.

Three of the four categories are tolerable, while only false negatives need to be viewed closely. Un-monitored failures at the cores or tasks neglect any reliability gain. However, in the case of the decision modules false negative allocations imply that a faulty task allocation is issued or an allocated task not discarded. Chapter 4.2 analyzed the failure susceptibility of the components of AAHS, clarifying the behavior of AAHS of false negative decisions of the decision modules. Summarizing, two distinctive, erratic monitor decisions are existing:

- Un-monitored failures at the cores or the tasks are devastating to the reliability of the complete system.
- Un-monitored failures at the decision modules are classified as Failure Class *B* and therefore no harm to the reliability of AAHS.

5

Design Methodology

Proven the feasibility, the design of the analog reliability-aware architecture follows up. As a spin-off the dependability analysis allows to generate the specifications needed for the different amplifiers, which are designed from scratch. However, Chapter 1.2 states the difficulty of designing, which are classified as Design Failures and Technology Effects, shown in Figure 1.2.

An automated design of analog circuits with knowledge of all critical areas, integrated as design rules, would minimize these design errors and increase reliability. Computer-aided design (CAD) tools help to partially automate the design flow, as shown in [Gie05]. The challenge [Rut06] is the manual design of the circuit structure, the structural synthesis is still mostly done manually. Though, the consecutive sizing and simulation steps are automated. The automatic sizing has evolved to industrial design flows through its integration into the design tools. First approaches [WH06, DV08] to automate those steps were developed using building blocks and symbolic analysis to generate topology synthesis of analog circuits. For the scope of this thesis a semi-automated synthesis framework [MMH11], presented in Chapter 5.2, is used to reduce the time consuming task of designing the different OpAmps and OTAs.

Figure 5.1 shows the methodology to fully design AAHS, from the algebraic models to synthesis to layout. The methodology starts by describing the system

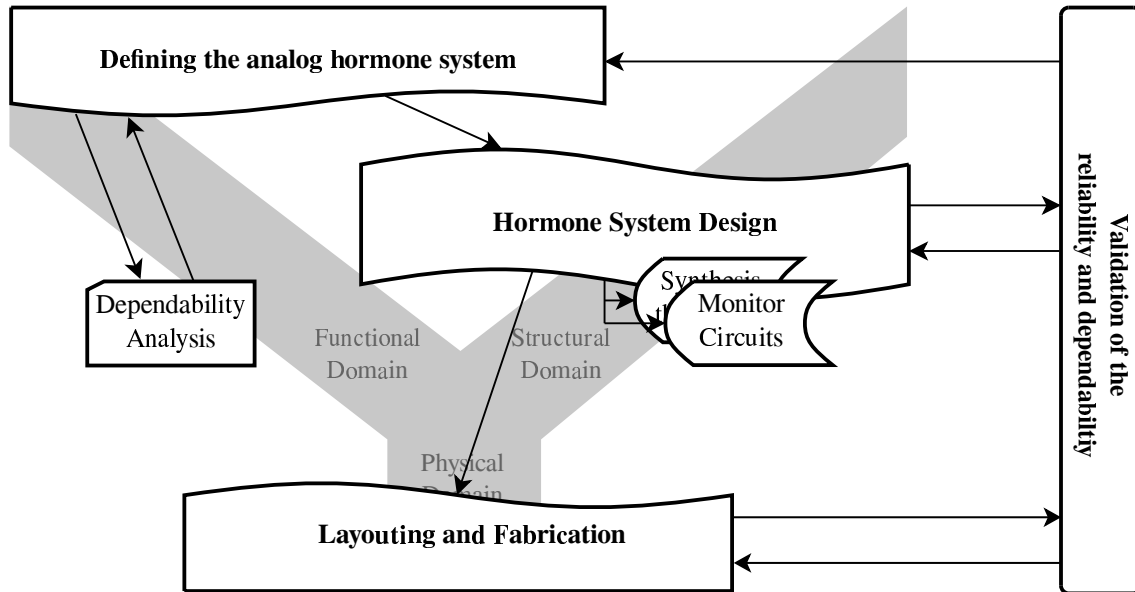


Figure 5.1: Flow of the Design Methodology of the Hormone System

and performing a dependability analysis to state the feasibility of the architecture. From the analysis derives a set of parameters, which are used to define the specification of all components of the system based on the chosen topology and the used technology. The design of the hormone system follows next, with the synthesis of the AAHS and the definition of the monitor circuit based on the reliability analysis. For the synthesis, the next steps are the circuit sizing and the layout generation, which eventually leads to the fabrication of the hormone system for real-world measurements and validation. The proposed methodology is a specialized design flow to create the highly dependable, analog hormone system as a mixed-signal task distribution system.

5.1 Specification Generation

The feasible set of all inequalities and equations (Chapter 4.1.1) defines the dependencies of the parameters of the hormone system. Those dependencies allow the derivation of the system specification and the specifications of the amplifiers to automate the system design. Figure 4.4 on page 90 shows the region, the polyhedron, of the feasible set, characterizing the relationship of those three parameters in regard to the fixed eager value E_γ . The values of the four variables with

fixed E_γ of Table 4.4 on page 88 derive the hormone values of the voltage- and the current-based architectures, as presented at Table 5.1. The hormone values are depending significantly upon the used process technology. For this thesis all components are designed using the AMS Design Hitkit v4.10 for a $0.35 \mu m$ bulk CMOS process with a supply voltage of $3.3V$, aligned to $V_{ref} = 1.65 V$ and maximum signal amplitude $V_{max} = \pm \frac{V_{DD}}{2} = \pm 1.65 V$. Thus, the maximum variation of the hormone values Δ_{max} aligned to the zero point is given at the last column of Table 5.1.

Table 5.1: *Derived Hormone Values*

	$\theta_{\gamma,i}$	α	β	E_γ	Δ_{max}
nominal:	0.0911	0.2722	0.0611	0.2424	± 1
as voltage:	$0.15 V$	$0.45 V$	$0.1 V$	$0.4 V$	$\pm 1.65 V$
as current:	to define	$2.722 \mu A$	$0.611 \mu A$	$2.424 \mu A$	$\pm 10 \mu A$

As stated on page 87, the robustness value $r_{CC} = \beta$, since $\theta_{\gamma,i} \geq 0.09$. Therefore, the column defining the β hormone values indicates the robustness of each approach, the fault-tolerance for example to noise and other voltage drifting effects:

- For the voltage-based architecture any noise and voltage variation less than $\beta = r_{CC} = 100 mV$ has no effect on the hormone system. Hence, it can be ignored.
- Choosing $1 \mu A$ as base unit for the current-based hormones is a trade-off between robustness and size of the OTAs, as Chapter 5.3.2 will show.

The voltage of the $\theta_{\gamma,i}$ of the current-based architecture has yet to be defined, since the Res. OTA determines the Input Voltage Range (IVR) of the Schmitt Trigger and therefore the threshold voltage to trigger accordingly. The IVR is defined by the technology and the range of the input voltage of a single differential amplifier stage before reaching saturation. Hence, the maximum IVR can be pessimistically defined by Equation (5.1) with $V_{DD} = 3.3 V$, $V_{sat} = 0.3 V$ and $V_{th} = 0.7 V$.

$$\begin{aligned} \max(IVR) &\leq V_{DD} - 2(V_{th} + V_{sat}) \\ &\leq 1.3 V \end{aligned} \quad (5.1)$$

For the current-based architecture the operational voltage region is set to $V_{\text{ref}} \pm 1 V$ (the nominal hormone value), indicating a second set of voltage hormone value. As example, the threshold of the ST for the current-based architecture is set to $\theta_{\gamma,i} = 0.0911 V$ for the further specification generation of the OTAs. Due to the shrunken operational voltage region the robustness to noise and similar effects decreases to $61 mV$.

Next, to generate the specifications of the two OpAmps (Local and Global Adder), three OTAs (Shunt, Measure and Res. OTA) and the two STs, the stability constraint of AAHS has to be defined. Recalling the stability constraint $\tau_{\text{stable},i}$ of equation (3.4) on page 60 the slew rates of the components of the decision unit need to be faster than the global hormone loop $\tau_{G,i}$. A desired task allocation time of less than $0.25 \mu s$ indicates a minimum slew rate SR_{min} of $13.2 \frac{V}{\mu s}$ as Equation (5.2) shows.

$$\begin{aligned} SR_{\text{min}} &= \frac{V_{\text{DD}}}{\tau_{L,\gamma,i}} \\ &= \frac{3.3 V}{0.25 \mu s} \end{aligned} \quad (5.2)$$

However, to ensure a task reallocation process (including a previous task drop) in less than $0.25 \mu s$, the slew rates of the Schmitt Trigger must be more than twice as fast, changing SR_{min} to at least $27.5 \frac{V}{\mu s}$. The slew rate of the Schmitt Trigger must be faster than SR_{min} , closely followed by the slew rate of the Local Adder. The slew rate of the Global Adder only needs to be slower than the slew rate of the Local Adder to fulfill the stability constraint $\tau_{\text{stable},i}$ as quickly as possible. For the OTAs the slew rates dependencies differ slightly. The Res. OTA converts the local current hormone level into a voltage for the Schmitt Trigger, implying the need to be as fast as the Schmitt Trigger. A similar relation applies to the Shunt OTA and the Measure OTA, their slew rates should also resemble, but slower than SR_{min} . Further, a set of additional variables are defined for the specification generation:

- Safety margin $s_m = 2.5$,
- Factor for all resistors $R_f = 25$,
- $V_{\text{SS}} = 0 V$,
- Accuracy parameter $\epsilon(V) = 40 mV$, respectively $\epsilon(A) = 0.244 \mu A$.

With those preliminary variables the specifications of the different amplifiers are generated according to the formulas of the two Tables 5.2(a) and 5.2(b). In

Table 5.2(a) the specifications for the Global Adder, the Local Adder and the ST are derived, while Table 5.2(b) focuses on the OTAs. The following specifications are calculated:

OpAmps	OTA
Gain	G_m
R_{Load}	R_{Load}
C_{Load}	C_{Load}
Overshoot	-
-	Input Voltage Range (IVR)
Output Voltage Range (OVR)	Output Resistance (OR)
Offset	Offset
Slew rate (SR)	Slew rate

Further, the AMS process technology specifies the bias source, which supplies $11.45 \mu A$. Accordingly all amplifiers and current mirrors are sized. The generated specifications of the two OPs, three OTAs and two STs are listed in Table 5.3(a) and 5.3(b).

5.2 Semi-Automated Analog Circuit Design

In order to fully synthesize the proposed architecture from specification to layout a semi-automated analog synthesis framework [MMH11, MH14] is used. To achieve the envisioned specification mandatory for the technical feasibility of the architecture the needed operational amplifiers are design from scratch. This extremely time consuming task is superseded by the synthesis framework, which provides for the specification defined process nodes fully sized, transistor level circuits. Figure 5.2 shows the design flow of the synthesis framework.

The objective of the framework is to generated circuits satisfying the given specification of AAHS. The framework needs three categories of inputs, while only the first category, the specification, varies according the demands of AAHS, the other two are untouched. [MMLH12] defines the three categories:

1. The wanted specification and the needed testbenches.
2. The circuit template and a library of abstract basic blocks.
3. The synthesis properties as a set of rules.

Table 5.2: Generating the Specification of the (a) OpAmps and (b) OTAs

(a)	Global Adder	Local Adder	Schmitt Trigger
Gain(dB)	$\geq 20 \log_{10} \left(\frac{V_{\text{ref}}}{2\epsilon(V)} \right)$	$\geq 20 \log_{10} \left(\frac{V_{\text{ref}}}{2\epsilon(V)} \right)$	$\geq 20 \log_{10} \left(\frac{V_{\text{ref}} \cdot sm}{2\theta_{\gamma,i}} \right)$
$R_{\text{Load}}(\Omega)$	$\geq \frac{10kR_f}{s_m N}$	$\geq \frac{10kR_f}{sm}$	$\geq \frac{10kR_f}{(\alpha(V)+\beta(V)) sm}$
$C_{\text{Load}}(F)$	$\leq s_m N C_0$	$\leq s_m C_1$	-
Over-shoot(%)	$\leq \frac{100 \cdot \beta(V)}{V_{\text{DD}}} \cdot 0.1e^{-1}$	$\leq \frac{100 \cdot \beta(V)}{V_{\text{DD}}} \cdot 0.1e^{-1}$	≤ 0.5
OVR(V)	$\geq 4(\alpha(V) + \epsilon(V))$	$\geq 4(\alpha(V) + \epsilon(V))$	$\geq V_{\text{DD}} - V_{th}$
Offset(V)	$\leq \epsilon(V)$	$\leq \epsilon(V)$	$\leq \epsilon(V)$
SR($\frac{V}{\mu s}$)	$\leq \frac{V_{\text{DD}}}{\tau_{L,\gamma,i}}$	$\geq \frac{V_{\text{DD}}}{\tau_{L,\gamma,i}}$	$\geq \frac{V_{\text{DD}}}{\tau_{L,\gamma,i}}$
(b)	Shunt OTA	Measure OTA	Res. OTA
$G_m(S)$	$\frac{10\alpha(I)}{s_m \text{IVR}[1]} \pm 10\%$	$\frac{10\alpha(I)}{s_m \text{IVR}[2]} \pm 10\%$	$\frac{\beta(I)+E_{\gamma}(I)-\epsilon(I)}{\theta_{\gamma,i}(V)} \pm 10\%$
$R_{\text{Load}}(\Omega)$	equiv. to Res. OTA	$\geq \frac{s_m \text{IVR}[2]}{10\alpha(I)}$	$\geq \frac{1}{G_m[3]}$
$C_{\text{Load}}(F)$	$\leq N C_{\text{load}}$ of Diff. Pair(Measure OTA)	equiv. to Res. OTA	$\leq C_{\text{load}}$ of Diff. Pair(ST)
IVR(Ω)	$\geq 2(2\alpha(V) + \epsilon(V))$	$\geq 2(2\alpha(V) + \epsilon(V))$	$\geq 2(2\alpha(V) + \epsilon(V))$
OR(Ω)	equiv. to Res. OTA	$\geq \frac{30\beta(V)}{\epsilon(V) \frac{10\alpha(I)}{s_m \text{IVR}[2]}}$	$\geq \frac{30\beta(V)}{\epsilon(V) G_m[3]}$
Offset(V)	$\leq \frac{100\epsilon(V) G_m[3]}{G_m[1]}$	equiv. to Shunt OTA	$\leq \frac{100\epsilon(V) G_m[3]}{G_m[3]}$
SR($\frac{V}{\mu s}$)	$\leq \frac{V_{\text{DD}}}{\tau_{L,\gamma,i}}$	equiv. to Shunt OTA	$\geq \frac{V_{\text{DD}}}{\tau_{L,\gamma,i}}$

Table 5.3: Set of Derived Specifications of the (a) OpAmps and (b) OTAs

(a)

	Global	Local	Schmitt Trigger	
	Adder	Adder	$\theta_{\gamma,i} = 150 mV$	$\theta_{\gamma,i} = 91 mV$
Gain	$\geq 26.2 dB$	$\geq 26.2 dB$	$\geq 27.09 dB$	$\geq 27.09 dB$
R_{Load}	$\geq 12.5 k\Omega$	$\geq 100 k\Omega$	$\geq 181.8 k\Omega$	$\geq 299.9 k\Omega$
C_{Load}	$\leq 10 fF$	$\leq 2.5 pF$	$\leq 1 pF$	$\leq 1 pF$
Overshoot	$\leq 0.03\%$	$\leq 0.03\%$	$\leq 0.5\%$	$\leq 0.5\%$
OVR	$\geq 1.95 V$	$\geq 1.95 V$	$\geq 2.6 V$	$\geq 2.6 V$
Offset	$\leq 40.3 mV$	$\leq 40.3 mV$	$\leq 40.3 mV$	$\leq 24.4 mV$
SR	$\leq 27.5 \frac{V}{\mu s}$	$\leq 27.5 \frac{V}{\mu s}$	$\geq 27.5 \frac{V}{\mu s}$	$\geq 27.5 \frac{V}{\mu s}$

(b)

	Shunt OTA	Measure OTA	Res. OTA
G_m	[8.891 μS , 10.75 μS]	[8.891 μS , 10.75 μS]	[15.643 μS , 18.928 μS]
R_{Load}	55.2 k Ω	102.25 k Ω	55.2 k Ω
C_{Load}	450 fF	500 fF	500 fF
IVR	$\geq 1.11 V$	$\geq 1.11 V$	$\geq 1.11 V$
Output - Resistance	$\geq 4.14 M\Omega$	$\geq 7.66 M\Omega$	$\geq 4.14 M\Omega$
Offset	$\leq 0.45 mV$	$\leq 0.45 mV$	$\leq 0.24 mV$
SR	$\leq 27.5 \frac{V}{\mu s}$	$\leq 27.5 \frac{V}{\mu s}$	$\geq 27.5 \frac{V}{\mu s}$

The performances (for example gain, slew rate, area, offset, overshoot) are evaluated in SPICE accuracy. The circuit template defines the top level structure of the circuit. The abstract basic blocks represent elementary, analog building blocks, which are used for the latter evaluation. The synthesis enables the fine tuning of the topology generation, while the maximum block count controls the process, enclose the design space and in some extend affect the amount of the resulting circuits.

The synthesize flow begins with the generation of new topologies based on the circuit template and the abstract basic blocks, enclosed by the repeatedly applied synthesis rules. The topologies not satisfying the input/output specification or exceeding the maximum block count are discarded, reducing the total amount of topologies significantly [MMLH12]. The remaining set of topologies are then expanded to substitute the abstract basic blocks with their circuit representative

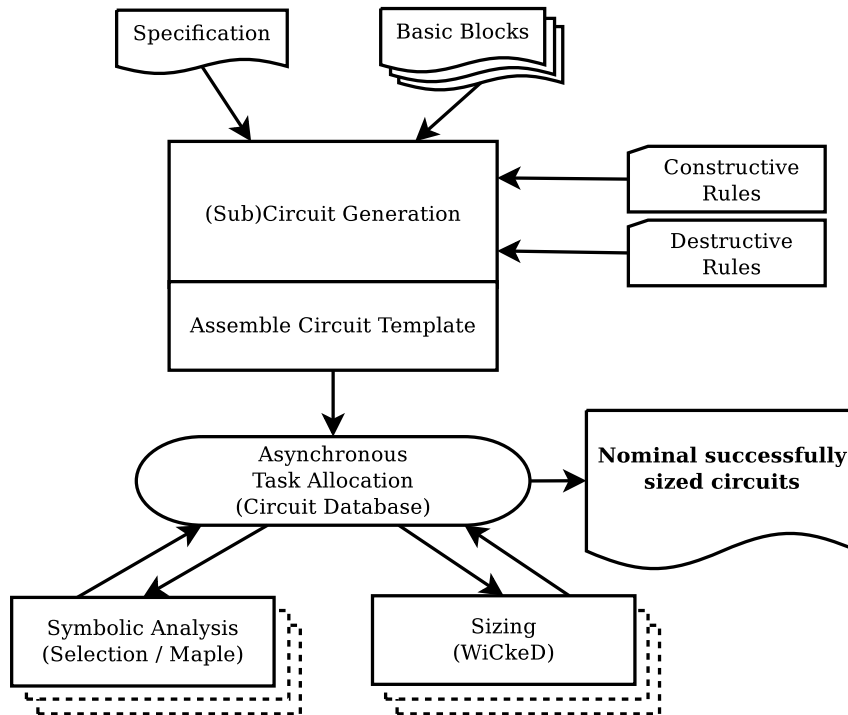


Figure 5.2: Fully Automated Analog Synthesis Framework Flow [MMLH12]

in an either symmetric or asymmetric manner [MMH11]. Symmetric expansion implies that equivalent abstract basic blocks are never substituted with differing analog building blocks. Contrary, each abstract basic block is expanded with all its variants, if the asymmetric expansion is chosen. After the generation process, the set of processed circuits is checked by an isomorphism algorithm, detailed described in [MMLH12], to filter all replica. The resulting set of unique circuits undergo a final sizing step [MMH11] and are nominal optimized in respect to the specification using a commercial tool [Mun].

5.3 Hormone System Design

The hormone system itself is designed to be completely symmetrically. The concept of redundant cores and a decentralized distribution system require the symmetry of the architecture. However, the symmetry also implies that the tasks each core applies to is considered with equal eagerness E_γ . A core with a perfectly symmetric decision unit could simultaneously apply multiple tasks, followed by

a discard due to the E_γ Switch. This process is constantly repeated, if not intersected beforehand.

An order of importance of tasks, which tasks are to be taken first, second, third and so on, demands an asymmetry within the decision units. Therefore, next to the symmetric concept of the decision units of the redundant cores, marginal asymmetries within two components enable orders of importance and solve the oscillating behavior of a core applying for multiple tasks:

1. Asymmetric designed E_γ Switches,
2. Two differently designed Schmitt Trigger.

5.3.1 Hormone System Synthesis

The synthesis runs of the semi-automated synthesis framework were executed with a two socket Quad-Core Intel Xeon E5520, each core double threading, and 36 GB RAM. Each run was done within five hours and provided for each given specification (Table 5.3(a) and 5.3(b)) a wide range of usable amplifiers for the two methodologies. The accomplished performances of the synthesized amplifiers are shown in Table 5.4(a) and 5.4(b) respectively. During the synthesis runs and viewing the wide range of usable amplifiers following observations are made:

Final Selection: The deciding factor for the finally chosen circuits is the smallest offset. An exception is the circuit to be used as Schmitt Trigger, which is chosen due to the highest almost symmetrical slew rate.

Challenges: The most challenging performances specified are the low overshoot to prevent spurious switching and the low load resistances of the circuit used as Global Adder.

Special Features: For the measurement of the OTAs the IVR and the output resistance is of importance, while the overshoots and the OVR are negligible. All circuits are optimized in terms of needed area.

Special Relations: The G_m of the Shunt OTA and the Measure OTA need to be almost identical, as do the slew rates of the Res. OTA and the ST. The G_m of the Shunt OTA determines the current value of Global Hormone Level, while locally the G_m of the Measure OTA provides the current α value. The slew rate of the Res. OTA needs to adapt to the slew rate of the ST to avoid faulty trigger.

Table 5.4: Measurement Table of the Semi-Automated Synthesized (a) OpAmps and (b) OTAs [vRMH15]

(a)

OP Performances	Global Adder	Local Adder	Schmitt Trigger
Gain	35.2 dB	27.3 dB	41.13 dB
Overshoot falling	0.017%	0.015%	0.18%
Overshoot rising	0.009%	0.027%	0.19%
OVR	2.48 V	2.6 V	2.64 V
Offset	36.6 μV	-2.06 μV	-17.37 μV
Slew rate falling	14.1 $\frac{V}{\mu s}$	74.78 $\frac{V}{\mu s}$	89.54 $\frac{V}{\mu s}$
Slew rate rising	13.6 $\frac{V}{\mu s}$	67.45 $\frac{V}{\mu s}$	82.22 $\frac{V}{\mu s}$
Phase margin	70.8°	66.1°	41.8°
Power	1.16 mA	0.31 mA	0.6 mA

(b)

OTA Performances	Shunt OTA	Measure OTA	Res. OTA
G_m	9.76 μS	10.03 μS	18.09 μS
IVR	1.17 V	1.18 V	1.4 V
Output Resistance	56.23 M Ω	40.02 M Ω	15.2 M Ω
Offset	0.25 mV	0.09 mV	-9.5 μV
Slew rate falling	0.4 $\frac{V}{\mu s}$	0.41 $\frac{V}{\mu s}$	80.9 $\frac{V}{\mu s}$
Slew rate rising	0.4 $\frac{V}{\mu s}$	0.42 $\frac{V}{\mu s}$	80.8 $\frac{V}{\mu s}$
Phase margin	84.4°	81.4°	68.9°
Power	0.24 mA	0.15 mA	0.8 mA

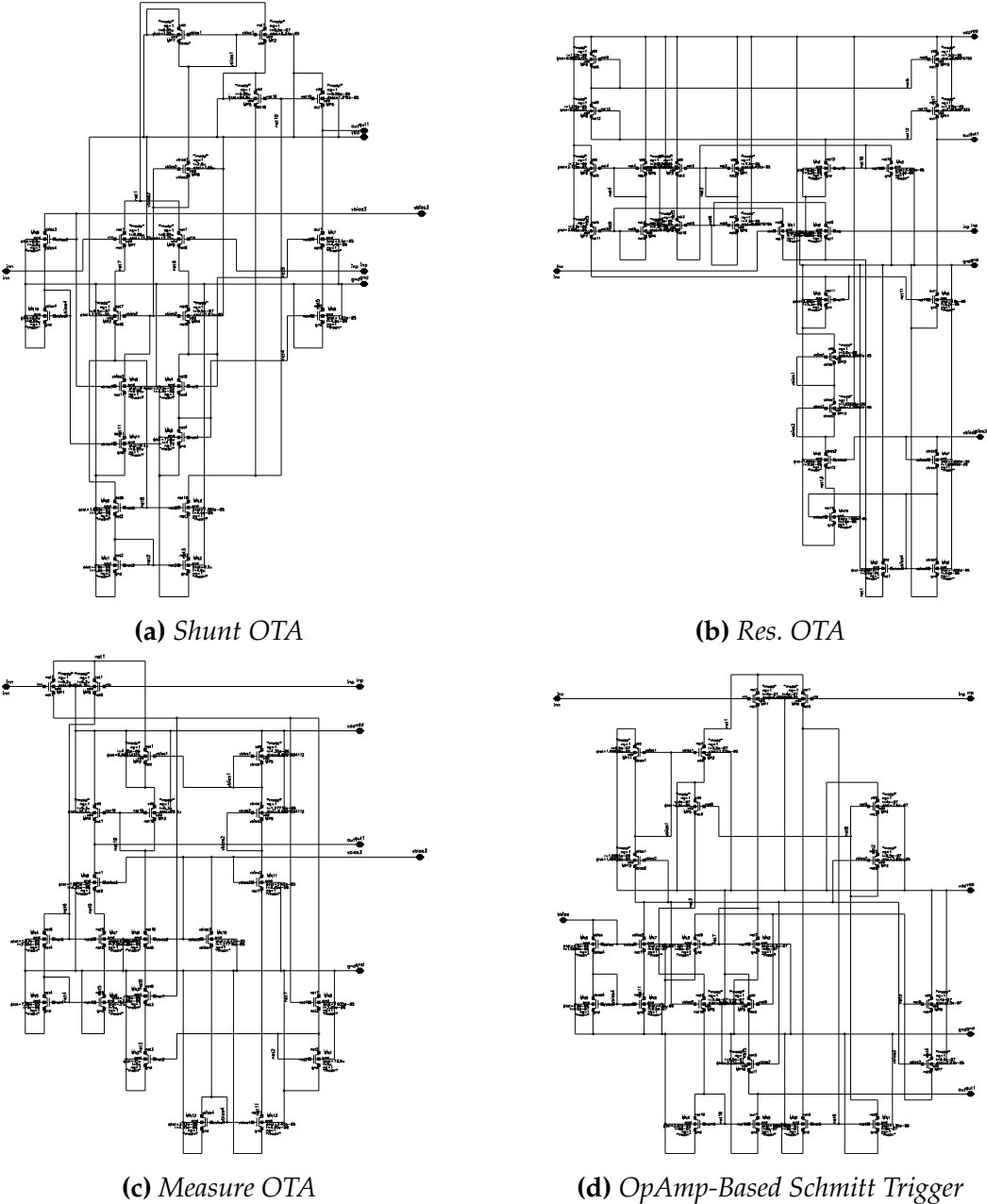
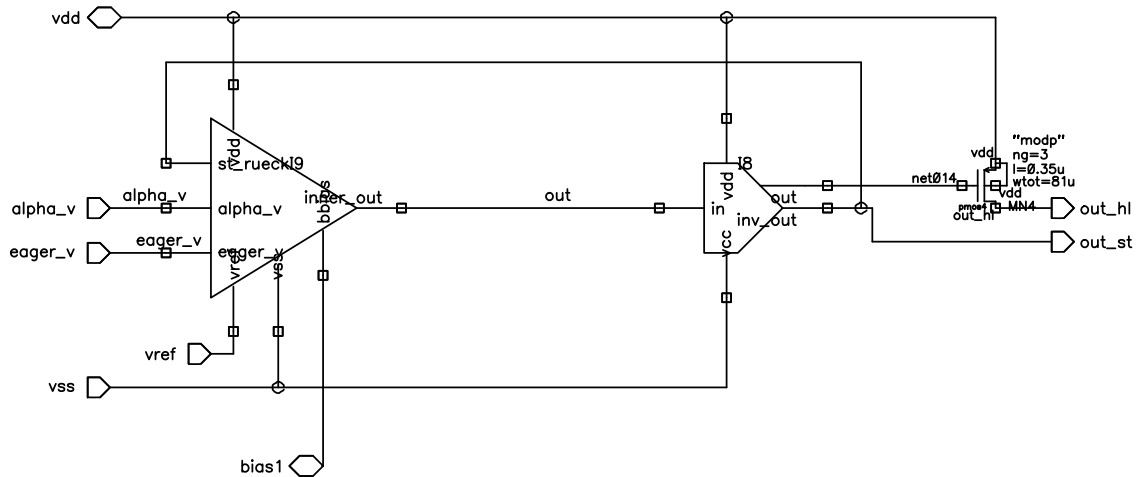
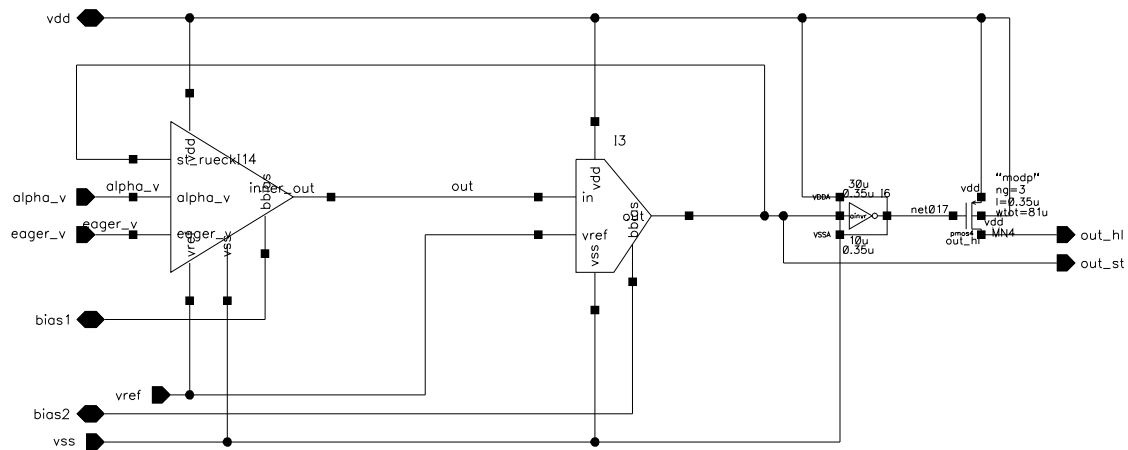


Figure 5.4: Schematics of the Semi-Automated Synthesized Amplifiers of OTAs and ST



(a) Schematic with a 6 Transistor ST



(b) Schematic with an OpAmp-based ST

Figure 5.5: Schematics of the Decision Modules using OpAmps [vRMH15]

OpAmp, which make is rather confusing to view and compare all internal signals. Yet, the output of the Schmitt Trigger is designed to be regarded as:

- V_{DD} implies the allocation of the task, which the decision module applied for,
- V_{SS} implies dropping the allocated task, else do nothing.

Due to the inverting character of the Global Adder, the inverted Schmitt Trigger output for the Global Hormone Level is needed to align the suppressor output between V_{ref} and V_{DD} . The scaling of the suppressor value to $\alpha(V)$ is done by the chosen resistors of the Global Adder. For the current-based architecture

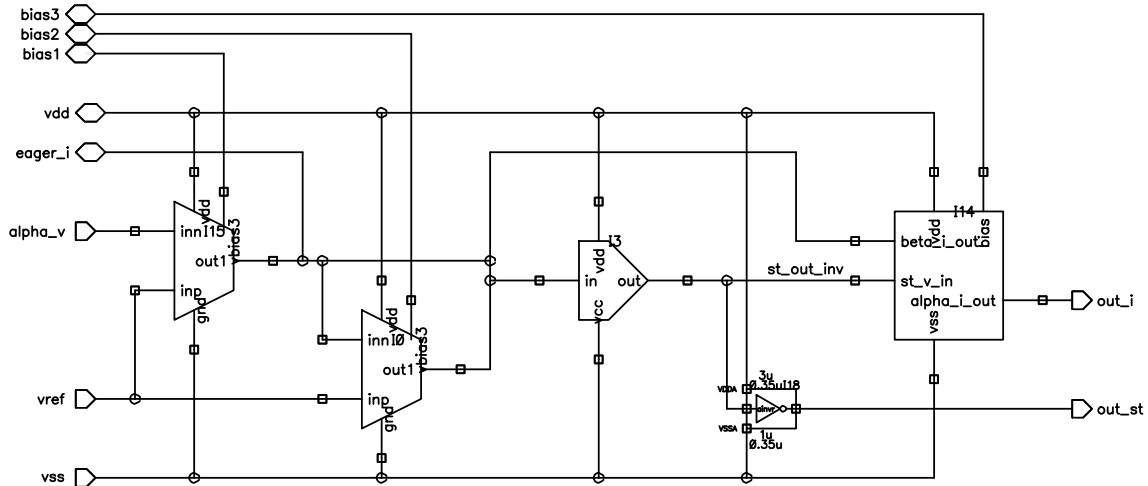


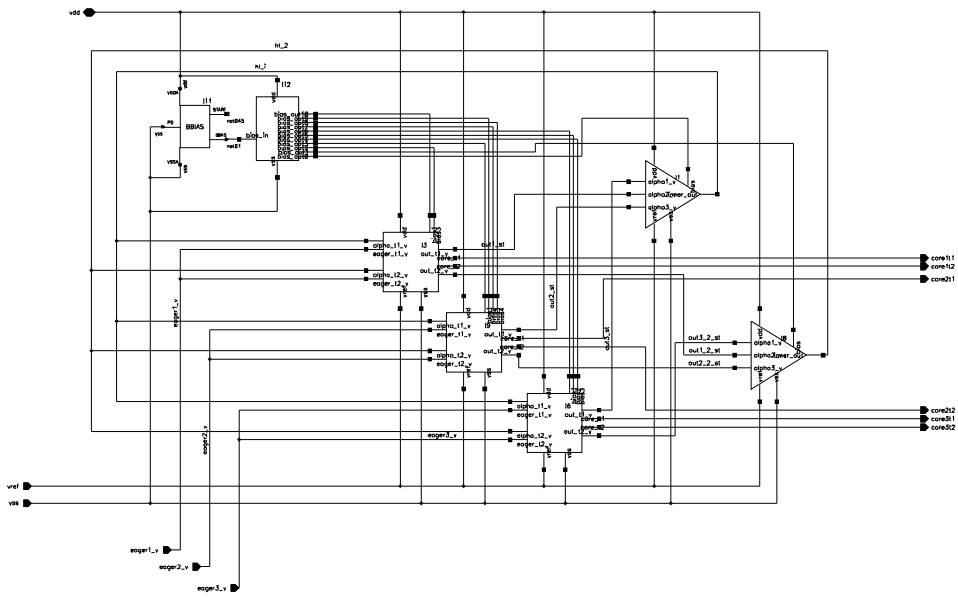
Figure 5.6: Schematic of the Decision Module using OTAs [vRMH15]

the inverted Schmitt Trigger output is needed to display V_{DD} as task allocating signal. The actual Schmitt Trigger output controls the $\alpha \beta$ -current mirror to apply the local accelerator and the global suppressor.

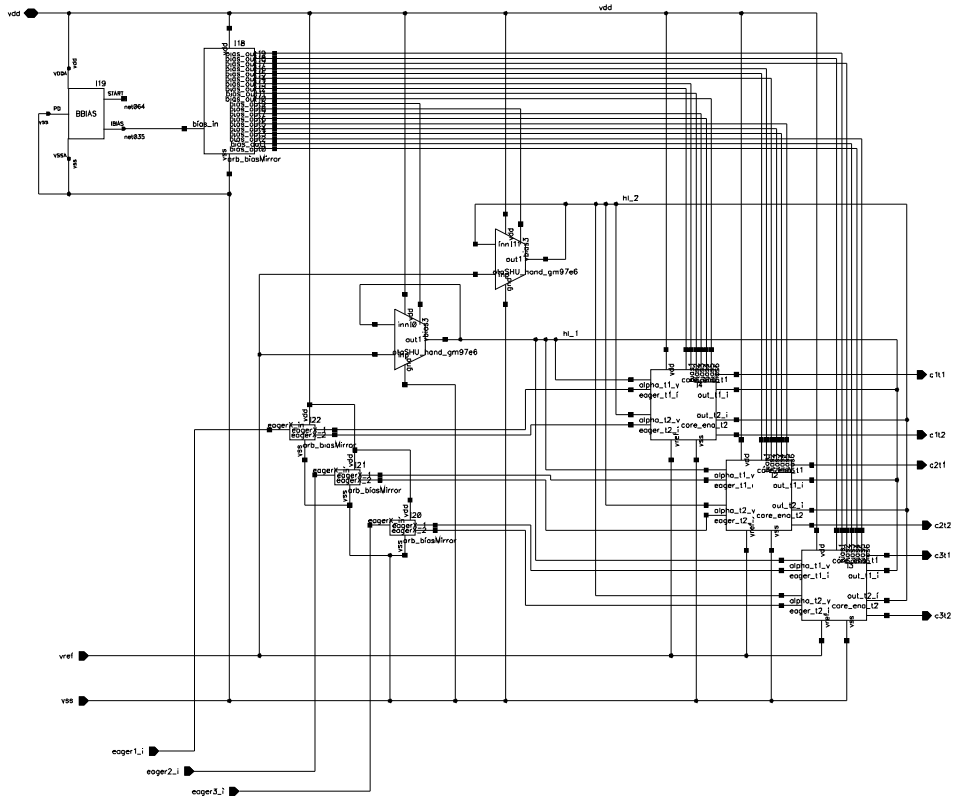
The schematics of the OpAmps (Figure 5.3) interconnected as voltage adder with a feedback loop and four resistors is spared within this thesis, since it is presumed to be fundamental knowledge. Also, this applies to any kind of current mirror. Finally, the schematics in Figure 5.7 show the two different AAHS architectures, (a) is realized by OpAmps and (b) shows the OTA-based approach. The results of the simulations of the two designed AAHS architectures can be seen in Chapter 6.1.1.

5.3.2 Hormone System Layout

Having the two AAHS architectures designed and running, the followup of the design process is laying out the two versions. Due to the manual layout process, a straightforward place and route methodology has been chosen. Furthermore, the relaxed timing constraints allow to focus on the accuracy of the analog signal processing. The two fully layouted architectures of three decision modules to apply for two tasks is illustrated in Figure 5.8. The current-based architecture realized with OTAs is shown on the left side of the image, while the right side shows the voltage-based approach. According to the enumeration given in Figure 5.8 the various components are:



(a) Schematic of AAHS Realised with OpAmps



(b) Schematic of AAHS Realised with OTAs

Figure 5.7: Schematics of the Architectures

1. Two decision modules of one core to apply for two tasks realized with OpAmps, containing two Local Adders and two Schmitt Triggers with the threshold voltage of $\theta_{\gamma,i} = 150 \text{ mV}$.
2. Two decision modules of one core to apply for two tasks realized with OTAs, containing two Measure OTAs, two Res. OTAs and two Schmitt Triggers with the threshold voltage of $\theta_{\gamma,i} = 91 \text{ mV}$.
3. The two Global Adders, one for each global hormone loop.
4. The two Shunt OTAs, one for each global hormone loop.
5. The two current mirrors to distribute the current of the bias sources, which are greyed out (property of AMS), to each amplifier.

The top of the layout shows 18 transmission gates, which frame the two AAHS architectures. Their use is to allow a switching between the two architectures. The two fully layouted architectures have been extracted to allow realistic simulations. The results of those simulation runs and the measurements of the silicon can be seen in Chapter 6.1. Details of the layouts can be found in Appendix A.3.

The two architectures, as presented in Figure 5.8, require a total of 14 OTAs, 8 OpAmps and 12 Schmitt Trigger, determined by:

OpAmp: 6 Local Adders, 2 Global Adders, and 6 Schmitt Trigger,
 OTA: 12 local OTAs, 2 global OTAs, and 6 Schmitt Trigger,

but neglecting the E_γ Switches and the Alpha & Beta CM of the OTA-based approach, due to their small size compared to the amplifiers. Hence, the estimation formula of the approximately needed area of AAHS presented in [vRSH⁺15] has to be edited to

$$\square_{\text{AAHS}} = \square_{\text{amp}} \cdot \left(m + \sum_1^N (m_\gamma D) \right), \quad (5.3)$$

with m representing the number of tasks and N the number of cores. D implies the number of needed amplifiers ($D_{\text{OpAmp}} = 12$ and $D_{\text{OTA}} = 18$) of the decision module and m_γ denotes the amount of tasks core γ applies to. The area estimation of an amplifier \square_{amp} is determined by the chosen processing technology.

With the two fully layouted AAHS architectures the estimation formula (5.3) can be superseded by the actual size measurement (the enumeration matches the enumeration of Figure 5.8):

1. The area of the decision unit consisting of two decision modules and the E_γ Switch built with OpAmps is $374.15 \mu\text{m} \cdot 170.8 \mu\text{m} = 0.064 \text{ mm}^2$,

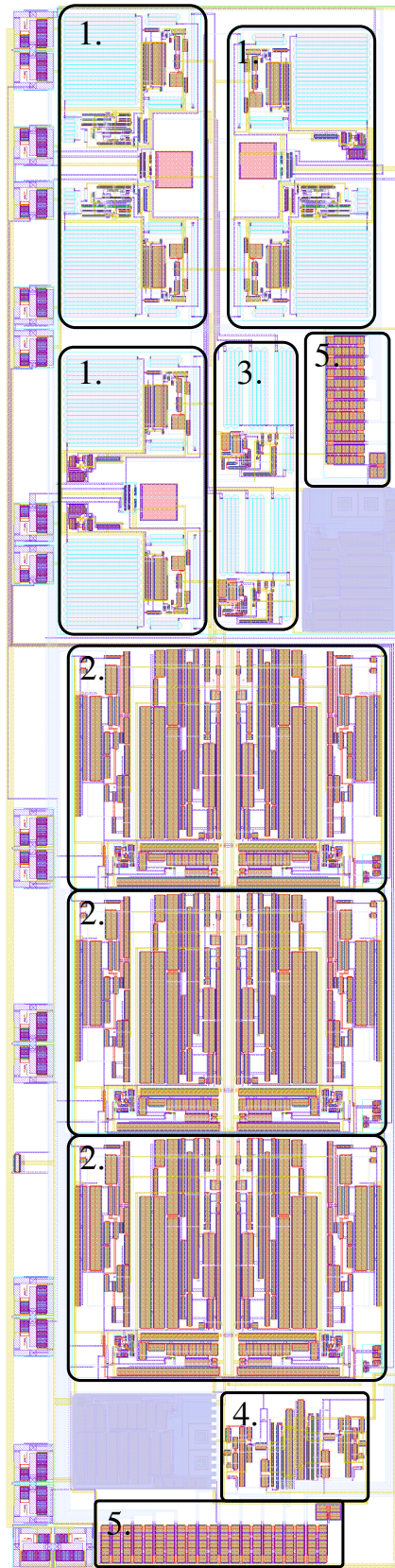


Figure 5.8: The Fully Laidout Architectures

2. The area of the decision unit consisting of two decision modules and the E_γ Switch built with OTAs is $372.9 \mu m \cdot 290.1 \mu m = 0.108 mm^2$,
3. The area of the Global Adder is $162.15 \mu m \cdot 92.85 \mu m = 0.015 mm^2$.
4. The area of the Shunt OTA is $101.05 \mu m \cdot 87.1 \mu m = 0.009 mm^2$,

With the size specified, the needed area of the architectures for any number of cores and tasks can be predicted. Recalling Chapter 3.2.2.2 a two-fold redundancy is proposed to eliminate the single Global Hormone Bus as single points of failure. For the area prediction of the voltage-based architecture, this simply implies to double the value of the Global Adder. The area prediction of the two-fold redundant current-based hormone bus is more complicated. Replicating the global Shunt OTA into the decision modules and doubling the global hormone bus lines implies to double the local Shunt OTA and Measure OTA for each task the core applies to also.

Contrary to previous predictions [vRBBH12, vRH12, vRSH⁺15], the voltage-based AAHS architecture requires slightly less area than the current-based approach. The size of the bias source is ignored, since it is highly depending on the chosen technology and the wanted current. Further, the area of the bias source has no relevance for AAHS itself.

5.3.3 Full System Task Migration

The distribution process using AAHS is done, if each task is allocated and the corresponding cores are activated. The decision modules of AAHS (the output of the Schmitt Trigger) are connected to the *select* input of the transmission gates. The selected transmission gates are connecting the corresponding cores to the data bus (see Figure 3.3 on page 57). Both transmission gates were sized to assure a current throughput of $100 \mu A$. If the decision modules of AAHS issue a task reallocation, the task transfer between the cores is done by the initial core being cut off from the data bus, while the transmission gates of the new core establish the connection to the data bus.

To determine the amount of needed transmission gates to transfer a task between different cores, the needed data bus connections of each task are of interest. For example, a PID-Controller has two input signals, but only one output. An extensive filter needs just one input signal, but three output signals (low-passed, band-passed and high-passed output signal). Equation (5.4) defines the amount

of needed transmission gates ($|TG|$) depending on the needed input and output signals ($|IN|$ & $|OUT|$) connected to the data bus.

$$|TG| = \sum_{i=1}^m \sum_{\gamma=1}^N (|IN_{\gamma,i}| + |OUT_{\gamma,i}|) \quad (5.4)$$

5.4 Monitoring

With the hormone loops fully designed, the last obstacle to assure self-control is self-reliance. This is achieved by locally monitoring the components of AAHS and the working cores. Chapter 4.2.3 gives an insight of the different failure modes in regard to AAHS and sample working cores. The hormone system itself is capable of handling any failure, which is classified in Failure Class C, no matter the failure mode. The other two failure classes demand monitoring with the focus on performance drifts exceeding the robustness value r_{CC} and non-recoverable component losses.

Failures of the working cores are not as easily handled. The two most basic failures of a working core are a failing core or a failing of the executed task. In either case, the monitor needs to set $E_{\gamma} = 0$ to prompt a task reallocation process. A further basic failure is the faulty execution of an allocated task. Monitoring the task execution to identify the failure demands a rather complex monitor. For example the monitor requires the reference output for comparison, as done for example to detect NBTI (Figure 1.5 on page 12) or TDDB (as presented in [NC13]).

The monitoring can either be online or a periodic offline monitor. For the offline monitoring a maintenance routine needs to be applied during which the monitored component is going down for the offline check. Hence, in regard to the continuous nature of the hormone system, online monitoring is to prefer. In any case, the monitors of the working core γ and the corresponding components of AAHS determine the eager value E_{γ} , with which the core participates at the task allocation processes.

5.4.1 Hormone System Monitoring

The Table 4.8 on page 103 states the most failure sensitive properties of several sample working cores, with the offset being the most dominant. Further, the lowest offset was the deciding factor to select the six amplifiers for the AAHS

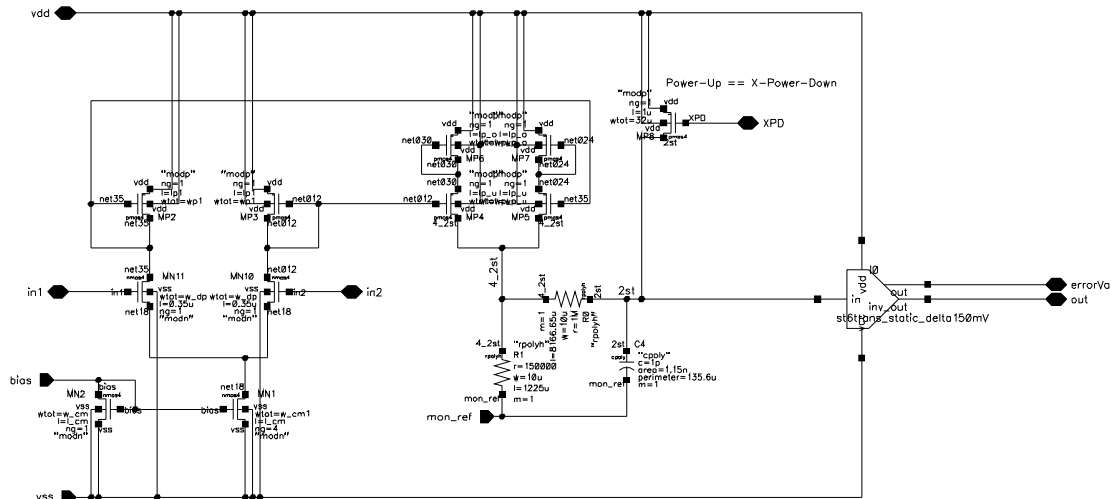


Figure 5.9: Voltage Drift Monitor

produced by the semi-automated design framework. This indicates that the offset is a dominant failure sensitive property for the components of AAHS as well.

Output voltage drifts less than the robustness value $r_{CC}(V)$ of any OpAmp will not affect the functionality of the voltage-based architecture. However, it should still be penalized by decreasing the eager value $E_{\gamma}(V)$. If the drift exceeds $r_{CC}(V)$, the eager value needs to be set to zero. The internal feedback loop of the negative input of the OpAmps allows to monitor the voltage drift of the output by comparing the two inputs. Such a monitor is not specialized on detecting HCI, NBTI or TDDB, but guards the general change done by a soft or severe failing effect:

- Aging transistors gradually vary the output voltage, an increasing offset decreases the eager value. Eventually, a task migration occurs, allowing to regenerate the aging effects.
- A destroyed transistor affects the performance of the OpAmp in two different ways:
 1. The output is massively affected and the voltage drift exceeds $r_{CC}(V)$, the voltage drift monitor sets $E_{\gamma}(V) = 0V$ and therefore catches the severe failing effect.
 2. With an output voltage drift of less than $r_{CC}(V)$, the OpAmp is still within the feasible region to guarantee dependability. The eager value

is adjusted, but no further actions are required, the hormone system configures and optimizes itself.¹

Figure 5.9 shows the schematic of a voltage drift monitor for any OpAmp used by AAHS. All current mirrors are symmetrical, except MN1 of the bias current mirror at the lower left. The symmetry is needed to measure the positive and negative difference of the differential pair. The monitor needs a reference voltage below the lower threshold voltage $-\theta_{\text{mon}}$ of the Schmitt Trigger. If the measured voltage difference of the differential pair exceeds the upper threshold, the trigger drops to zero, generating a health signal of zero. If the difference decreases below the lower threshold again, the trigger switches to *healthy*. The monitor is used to detect the offset drift ($\geq 100 \text{ mV}$) of the low-pass filter presented in Chapter 6.3. The occurring offset eventually exceeds 100 mV , the ST triggers and E_γ dropped.

5.4.2 Working Core Monitoring

Considering the Global Adder as a working core, as done in Table 4.8, the voltage drift monitor of Figure 5.9 is also applicable to monitor the offset of OpAmps used by the working cores. Such cores are, for example, low-pass filters and voltage adders.

In cases of monitoring the voltage difference of the input to the output, a simple monitor is published in [vRSH⁺15, Ch. 6]. The presented monitor of an output stage detects the recoverable effect of NBTI due to aging and voltage shifts due to failing transistors. Figure 5.10 shows the block diagram of a monitor measuring the absolute input-output voltage difference of a standard class AB-output stage. The difference signal is then frequency shaped and controlled by a Schmitt Trigger to derive two clearly distinct cases, instead of a fast on and off health signal oscillation. This 1-bit health signal controls the eager value, as

- a *living* health signal (the Schmitt Trigger output equals V_{DD}) implies an eager value $E_\gamma = 0.2424$,
- a *dead* health signal (the Schmitt Trigger output equals V_{SS}) implies an eager value $E_\gamma = 0$.

To incorporate further distinctions of the health signal to adjust the eager value in a more fine grained manner, the Schmitt Trigger has to be replaced by a more sophisticated subcircuit.

¹The Table 5.1 on page 107 provides the calculated value of $r_{\text{CC}}(V)$ of the voltage-based architecture.

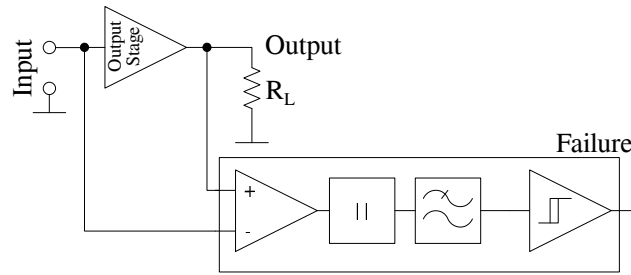


Figure 5.10: Block Diagram of a Monitor Circuit for an Output Stage [vRSH⁺15]

For the recoverable failures the monitoring indicates a noticeable increase in reliability of the working cores. Monitored errors or degrading effects lead to a decrease of the eager value, which eventually implies a task migration. The initially occupied core is powered down now, unstressed and recovering. After a predefined recovery phase, the core starts again and the monitor will notice the achieved degree of recovery, conceivably reactivating the eager value by setting $E_\gamma = 0.2424 \pm r_{CC}$. The decision module of AAHS rejoins the task allocation processes for the recovered working cores. However, a detailed analysis of the recovery and reactivation process is an interesting field of study of future work.

For autonomous robots, like vehicles or air-crafts, batteries are used to supply the appropriate voltage V_{DD} . A battery monitor checking the supply voltage of the full system is presented in Figure 5.11. The three LEDs visualize the state the battery is in:

- The *alive* LED states that the supply voltage is above 2.7 V (in regard to the chosen technology), classified as alive and working.
- The *good* LED states that the battery is full and $V_{DD} > 3.2 V$.
- The *dying* LED states that the battery is dying and $3.2 V > V_{DD} > 2.7 V$.

The three Zener diodes followed by the resistors and the three OpAmps at the left side of Figure 5.11 determine the voltage at which the LEDs come on. The top right amplifier connected to the switching transistor provides the change from *good* state to *dying*. The lower right OpAmp provides an eager value E_γ to manage the decision unit the monitor is connected to. Further, the circuit enables the distribution of an eager value, enhancing the system to self-control several batteries, charging the empty one, while the full batteries participate on allocating the task *supplying* V_{DD} .

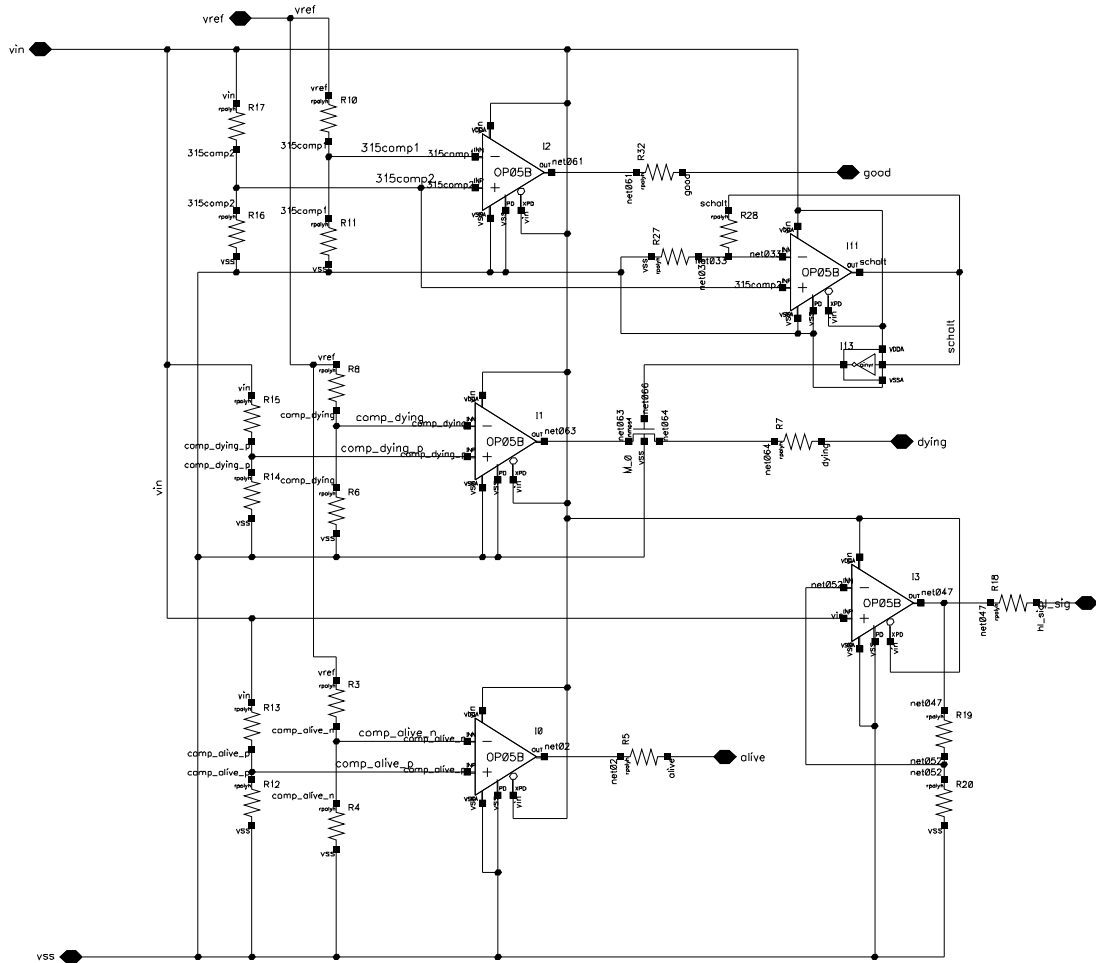


Figure 5.11: Monitoring the Supply Voltage of a Battery



Figure 5.12: Heartbeat Signal Monitor

5.4.3 Hormone Bus Monitoring

Monitoring the wires of the networks and the hormone buses increases the reliability of the proposed approach. Figure 5.12 introduces a simple heartbeat signal monitor. A sine is modeled periodically upon the different signals and is counted. However, the amplitude of the sine should be less than half the robustness value r_{CC} , not to interfere with the hormone values and AAHS. The oscillation can be counted, measured and the like by the heartbeat receiver, stating if the sine was transmitted correctly. For example, as illustrated in Figure 5.12, the transmitted sine has changed, delayed and with increased amplitude. Hence, the receiver monitor could be decreasing the eager values of the global unit assigned to that bus line. The result is that the affected bus line is not used anymore.

With redundant bus lines, as used by the two-fold Global Adder approach, the periodical sine of each line is compared to one another. As soon as the periodical heartbeats are transmitted incorrectly, a defect on the wire is most likely to be existing. Similar applies, if the counted heartbeats of the two redundant wires of the hormone bus differ. While a defect of the wire within the decision module of AAHS occurs, the monitor decreases the eager value of the core accordingly, eventually leading to a task reallocation. An erroneous wire within the hormone bus leads to a transfer of the hormone transmission upon the second hormone bus wire, keeping AAHS operational.

6

Results

To confirm the envisioned objectives mentioned in Chapter 2.5, the designed architectures are validated and evaluated against the approaches of Table 2.3 on page 50 with equivalent criteria. The validation of AAHS is done in terms of simulation runs of the schematics, an extracted view and the measurements of a prototype chip. Further, the motivating examples of Chapter 1.4 are picked up to show the application usage of AAHS. For the two proposed architectures of AAHS

- the reliability and dependability,
- the speed of allocation in seconds,
- the power consumption and
- the eager value allocation bounds

can be shown in comparison to the measurements of the fabricated chips. The two architectures are compared to one another and benchmarked on AHS. The reliability gain, real-time bounds, size overhead and scalability of the summary of AAHS is also stated. A conclusion of the results of the validation is done in Chapter 6.2.

6.1 Validating the Design

The validation of the correct functionality of the analog hormone system is based on the implementation with three cores and their six decision modules applying for two tasks. Due to the accordingly defined feasibility equations, the validation results and their complex behaviors can be projected upon any N cores and m tasks. However, the complexity of the distribution and the validation of the system state quickly rises.

With a system of three cores, eight different conditions exist describing the state of the cores, if they are either on (**t**) or off (**f**). For Table 6.1 it applies that $N = 3 : C_\gamma, C_\kappa, C_l \in 1..N$. Therefore, the amount of states s_j of a multi-core system is defined by $|s| = 2^N$. However, not every system state is tolerable. More

Table 6.1: Classification of the System States

System state s_j	Actual core state			Satisfied system state for		
	C_γ	C_κ	C_l	$m = 1$	$m = 2$	$m = 3$
s_1	f	f	f	false	false	false
s_2	f	f	t	true	false	false
s_3	f	t	f	true	false	false
s_4	f	t	t	false	true	false
s_5	t	f	f	true	false	false
s_6	t	f	t	false	true	false
s_7	t	t	f	false	true	false
s_8	t	t	t	false	false	true

so, the actual satisfying states differ, depending on the amount of tasks m to be executed by the system, which is also seen in Table 6.1. Further, a state coverage is given by equation (6.1):

$$C_{0,N,m} = \frac{\text{number of accepting states}}{\text{number of all states}}. \quad (6.1)$$

For the three cases ($m = 1, m = 2, m = 3$) the satisfying system states are:

Case m=1: A system with $m = 1$ is validated as correct, if the system states s_2, s_3 and s_5 are reached. All other system states presume at least a faulty allocation process. It follows that 3 of 8' system states with three possible distributions are satisfying, $C_{0,3,1} = \frac{3}{8}$.

Case m=2: A system with $m = 2$ is validated as correct, if the system states s_4 , s_6 and s_7 are reached. Again, all other system states presume a faulty allocation process. Further, it follows that ,3 of $8'$ system states are satisfying, $\mathcal{C}_{0,3,2} = \frac{3}{8}$, but with six distribution possibilities.

Case m=3: A system with $m = 3$ is validated as correct, if the system state s_8 is reached. Again, all other system states presume a faulty allocation process. Only ,1 of $8'$ system states is satisfying, $\mathcal{C}_{0,3,3} = \frac{1}{8}$. However, six distribution possibilities exist.

For a system with three cores the system and the satisfying states can be classified by a table, but for any $N \geq 4$ the hand written approach is getting error-prone. Equations (6.2) to (6.4) represent a formal description of the satisfiability condition of the system states to ensure the correct functionality of AAHS. The length of a word ω is denoted as $|\omega|$.

$$A(\omega) = \text{the number of elements } \mathbf{t} \text{ in } \omega, \quad (6.2)$$

$$\Omega_{AAHS}^N = \{\omega \in \{\mathbf{t}, \mathbf{f}\}^* \mid A(\omega) = m, |\omega| = N\}, \quad (6.3)$$

$$s_j = \begin{cases} \text{true,} & \text{if the core state assignment } \{C_1..C_N\} \in \Omega_{AAHS}^N \\ \text{false,} & \text{otherwise.} \end{cases} \quad (6.4)$$

Based on the three cases above, the following three observations are derived, respectively confirmed:

1. To distribute m tasks on N cores, $\frac{N!}{(N-m)!}$ different solutions are existing.
2. The constraint $m < N$ on page 63 leads to more accepting conditions and should be assured. However, in particular occasions $m = N$ is tolerable, for example to keep the system operating, until a dysfunctional or failed core is replaced and $m < N$ restored.
3. The state coverage is not a sufficient measure to state the optimal amount of tasks for N cores.

Therefore, the allocation coverage \mathcal{C}_1 , defined in equation (6.5), presents a better comparable measurement than the state coverage \mathcal{C}_0 , since it takes the amount of tasks m into account.

$$\mathcal{C}_{1,N,m} = \frac{\sum_{i=1}^m \frac{i}{m} |\Omega_{AAHS}^N|}{|S|} \quad (6.5)$$

Next, all eight system states have to be validated. The validation is based on the simulation of the schematic and an extracted view, and the measurements done of the prototype chip. Following conditions have to be proven:

Condition No-Allocation: State s_1 can occur only, if all cores are taken out of the allocation process. Otherwise by the definition of the Global Hormone Level G_i on page 58 the stability constraint hinders to reach this state in a stable manner.

Condition Double-Allocation: A multiple allocation of a single task can only occur at state s_4, s_6, s_7, s_8 . If the simulation results prove that a double allocation of a task is denied by the Global Hormone Level G_i and a restart of the allocation process is issued, the denial applies to triple, quadruple and more allocations also.

Condition Allocating Two Tasks: Allocating two tasks (**Case m=2**) requires the system to be in state s_4, s_6 or s_7 , while the states s_2, s_3, s_5 indicate that the priority task is allocated already. Therefore, three system states imply the optimal allocation case, while three other system states allocate the prioritized task, done by the marginal asymmetry of the E_γ Switches. However, the symmetric structure of the decision modules assures that any task, if a failure occurs, is reallocated, no matter the priority list. For the allocation coverage \mathcal{C}_1 evaluate to

$$\mathcal{C}_{1,3,2} = \frac{3 + \frac{1}{2} \cdot 3}{8} = 56.25\%. \quad (6.6)$$

Condition Allocating Three Tasks: Proving the last system state (state s_8) is done by the allocation of three tasks by three cores (**Case m=3**). Similar to allocating two tasks, the allocation coverage \mathcal{C}_1 is given with

$$\mathcal{C}_{1,3,3} = \frac{\frac{3}{3} \cdot 1 + \frac{2}{3} \cdot 3 + \frac{1}{3} \cdot 3}{8} = 50\%, \quad (6.7)$$

strengthening the second observation on the previous page. However, the prove of this condition is assumed by proving the rightness of allocating two tasks.

For **Case m=1** the allocation coverage is still $\mathcal{C}_{1,3,1} = \frac{3}{8} = 37.5\%$. Calculating the allocation coverage for $N = 4$ using $m = 1..N$ tasks, the highest value is received for $m = 3$ with $\mathcal{C}_{1,4,3} = 50\%$. This applies to $N = 5$ and $m \leq N$ tasks

also. Concluding, the best allocation coverage is achieved for any N cores with $m = N - 1$ tasks.

To validate the robustness of AAHS, the amplifiers within their testbenches and the whole hormone system are simulated against all process corners, which had no impact on the task distribution and its functionality. Hence, the simulations results against all process corners are negligible and not shown.

6.1.1 Simulation Runs of the Architectures

Voltage-based Architecture Figure 6.1 shows a simulation run of AAHS distributing two tasks on three cores using the voltage-based architecture. All cores are disabled at start-up by eager values of $0 V$ aligned to V_{ref} (see page 66). For the voltage-based architecture, task T_1 is regarded as the prioritized task.

After $1.1 \mu s$ the eager value E_3 begins to rise, eager value E_2 follows $0.2875 \mu s$ later. All three eager values rise with equal slope over $4 \mu s$ to $2.05 V$ corresponding to an eager value of $400 mV$. The cores C_3 and C_2 are beginning the allocation process once an eager value of $1.83 V$ is surpassed.¹ So far, the allocation is hindered by the global suppressor, as the feasible region is still not reached. After $4.081792 \mu s$ the eager value $E_2 = 1.91943 V$ and has still not reached the feasible region, even though hesitating by approximately $50 ns$ core C_2 drops task T_1 again. $0.3 \mu s$ afterwards, as $E_2 = 1.95022 V$ task T_1 is allocated successful.

The marker M1 (see Table 6.2 for the signal values) states the eager value E_3 leading to the successful allocation of T_2 . The value of $1.75626 V$ for the G_2 results from the inner loop of the decision unit of core C_3 , showing the depending influence of all loops of this architecture. The E_γ Switch is causing the mutual start-up allocation process of the cores to each task with insufficient eager values. As a core can not hold on to one task, the eager value is allowed through to the other decision unit of the core again, initiating the allocation process of the other task.

The oscillating behavior of the allocation process is shown enlarged in Figure 6.2(a) and enables to retrace the following timing evaluations:

- The Schmitt Trigger of the decision units are deciding much faster to allocate tasks than the global hormone levels require to adjust to the new values:

(a) $\tau_{G,1}$ requires $97.34 ns$ to adjust to the new hormone level η_1 .

¹The core C_3 begins the allocation process for task T_1 with $E_3 = 1.83802 V$, which is reached at $2.980242 \mu s$, while C_2 follows with $E_2 = 1.83025 V$ at $3.190007 \mu s$.

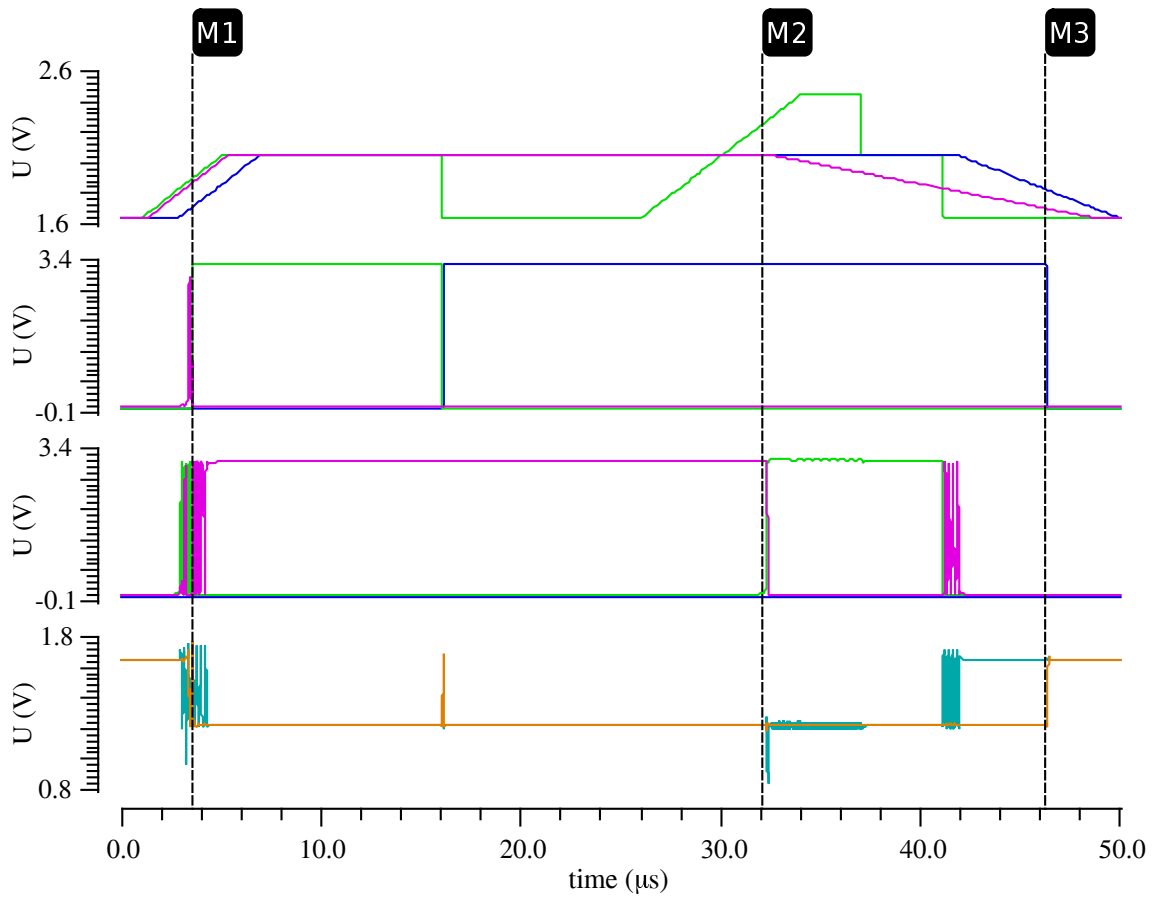


Figure 6.1: Simulation Run of the Voltage-Based Architecture

- **Strip 1** displays the eager values E_1 (blue), E_2 (violet), E_3 (green) measured in V
- **Strip 2** displays the task allocation of T_2 by C_3 (green) and C_1 (blue)
- **Strip 3** displays the task allocation of T_1 by C_2 (violet)
- **Strip 4** displays the G_i of T_1 (cyan), T_2 (orange) measured in V
- For the markers M1, M2 and M3 the values of the signals are shown in Table 6.2

Table 6.2: Signals of the Voltage-Based Architecture at Significant Time Steps

	M1	M2	M3
time	3.605512 μs	32.29433 μs	46.37564 μs
E_1	1.71555 V	2.05 V	1.82872 V
E_2	1.8718 V	2.05 V	1.70686 V
E_3	1.90055 V	2.26943 V	1.65 V
C_1	175.025 μV	3.29978 V	3.30515 V
T_2 C_2	310.097 mV	44.9861 mV	45.0018 mV
C_3	3.18356 mV	173.417 μV	173.401 μV
C_1	165.984 μV	173.41 μV	176.364 μV
T_1 C_2	174.539 mV	3.11141 V	49.8967 mV
C_3	474.646 mV	46.5034 mV	44.9946 μV
G_1	1.66223 V	1.22218 V	1.64805 V
G_2	1.75626 V	1.22213 V	1.21961 V

(b) $\tau_{G,2}$ requires 117.9 ns to adjust to the new hormone level η_2 .

During the adaptation phases to the new Global Hormone Levels the other decision units base their decisions on wrong values.

- The sum of all $\tau_{G,i}$ determines τ_{AAHS} , which also defines the minimum self-configuration time.

$$\begin{aligned}\tau_{AAHS} &= \sum_{i=1}^m \tau_{G,i} \\ &= 215.29 \text{ ns}\end{aligned}\tag{6.8}$$

- The shortest adaptation phase determines the slew rate $SR_E = \frac{\Delta E_i}{\Delta \text{time}}$ of the eager values to reach the feasible region for the allocation process.

The following assumptions derive:

- Rising the eager value slower than the fastest adaption phase (less than 97.34 ns) leads to faulty allocations:
 - (a) The worst case scenario is a permanent oscillating allocation behavior.
 - (b) The task prioritization is ignored as seen in Figure 6.1 by allocating T_2 .

(c) Several allocation attempts are necessary until a successful allocation occurs.

- The faster the slew rate of the Global Adders, the quicker τ_{AAHS} can be reached and the better the real-time constraints. However, SR_E of the eager value to reach the feasible operating region increases as well.

The reallocation process, which is happening after $16.1 \mu s$, is due to the loss of E_3 , enlarged in Figure 6.2(b). $10.99 ns$ after the drop of E_3 , the task T_2 is discarded and another $51.59 ns$ thereafter the task is allocated by core C_1 , a total of $62.59 ns$.

A task steal is issued once an eager value rises above $2.26 V$, as the values of the marker M2 at Table 6.2 show. The process takes $65.73 ns$ from the steal to the task discard at the initial core. The eager value of the steal, as defined in equation (6.9), matches twice the lower bound of the eager value.

$$E_{\text{steal},\gamma} = V_{\text{ref}} + 2(E_\gamma - r_{CC}) = 1.65 V + 2 \cdot 0.3 V \quad (6.9)$$

The allocation attempts of core C_2 for task T_1 after $41.2 \mu s$ result from the eager value $E_2 \approx 1.83625 V$, being too low to hold on the task already.

Current-based Architecture Figure 6.3 shows the simulation run of AAHS using the current-based architecture. Even though the Global Hormone Level is now a current with an value of $2.722 \mu A$, Figure 6.3 and 6.7 show the inverted voltage output of the Shunt OTA.

The eager values of the three cores are increasing with equal slope to $2.424 \mu A$ (as defined in Table 5.1), only differing by the starting time. The increase of the eager value E_3 begins at $1.1 \mu s$, closely followed by E_2 $19.817 ns$ afterwards. E_1 begins at $1.3 \mu s$. At marker M1 (Table 6.3) C_3 starts to allocate T_2 due to the prioritization of the second task. The values of the marker M2 show that a task steal demands a rise of the eager value E_3 of almost twice its value. Also of interest, the values of marker M3, showing how far the eager values need to drop to discard a task. $77.6179 nA$ are approximately 3% of the determined eager value hormone value. The eager values during the allocation, the task steal and the discard suggest that the feasible region to operate in for the current-based architecture has been altered. The alteration is examined more closely at the simulation run of the extracted view and the hardware measurements of this architecture.

The timings to adjust the Global Hormone Level and of the task reallocation process are given in Table 6.4, supported by the enlarged sub-graphs of Figure 6.4. Therefore, the timing constraints for $\tau_{AAHS} = 355.3 ns$.

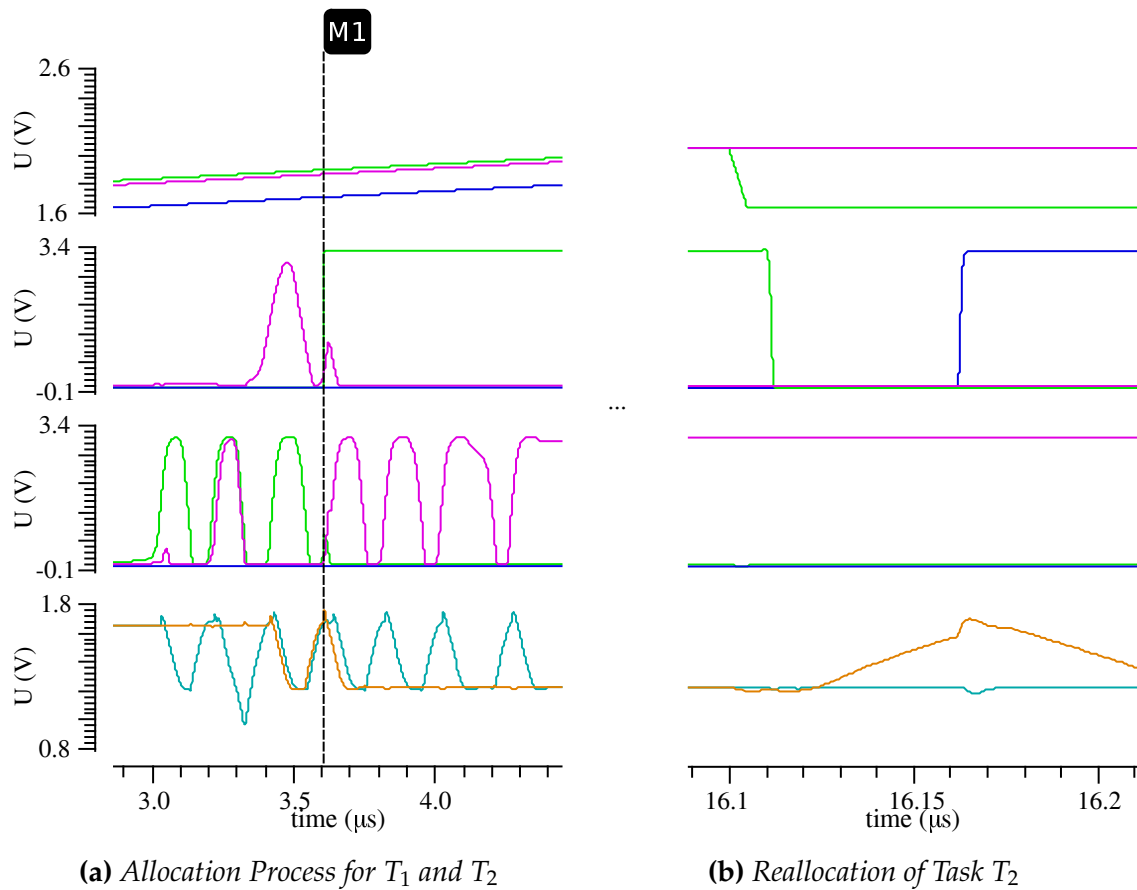


Figure 6.2: Allocation Processes of the Voltage-Based Architecture

- **Strip 1** displays the eager values E_1 (blue), E_2 (violet), E_3 (green) measured in V
- **Strip 2** displays the task allocation of T_2 by C_3 (green) and C_1 (blue)
- **Strip 3** displays the task allocation of T_1 by C_2 (violet)
- **Strip 4** displays the G_i of T_1 (cyan), T_2 (orange) measured in V

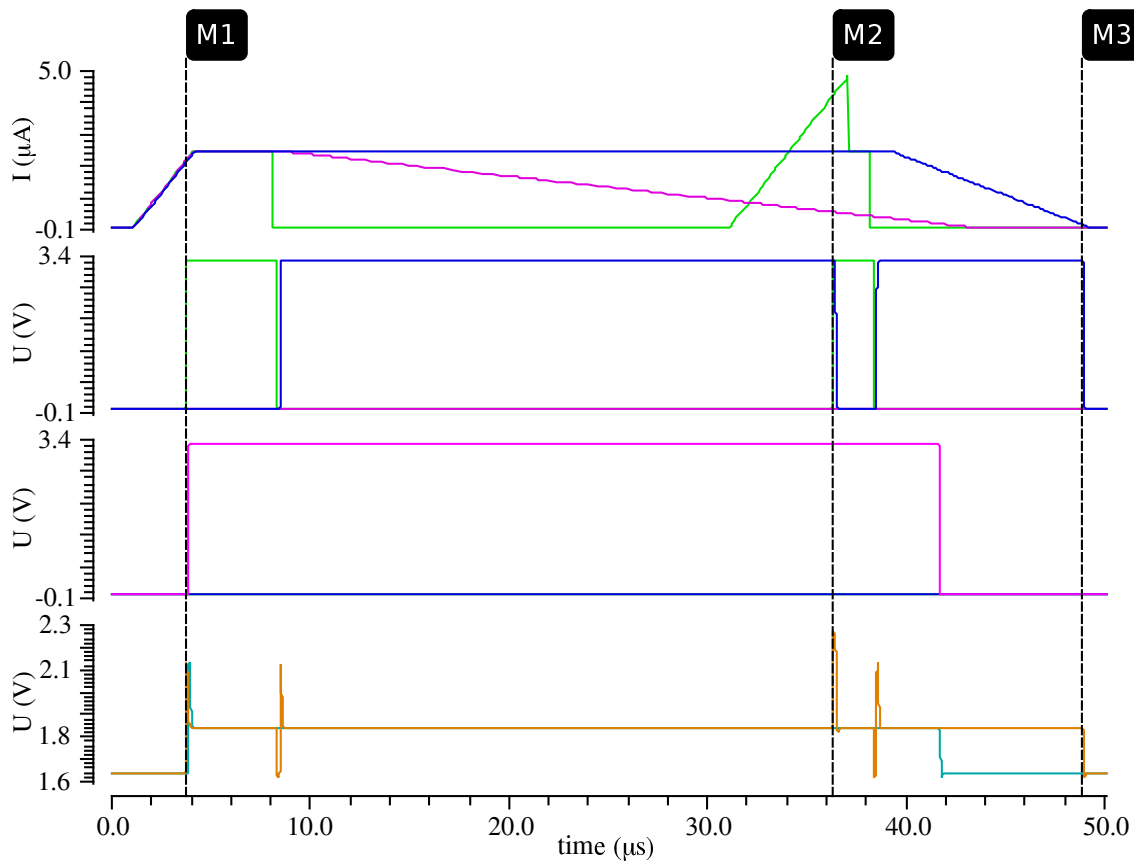


Figure 6.3: Simulation Run of the Current-Based Architecture

- **Strip 1** displays the eager values E_1 (blue), E_2 (violet), E_3 (green) measured in μA
- **Strip 2** displays the task allocation of T_2 by C_3 (green) and C_1 (blue)
- **Strip 3** displays the task allocation of T_1 by C_2 (violet)
- **Strip 4** displays the G_i of T_1 (cyan), T_2 (orange) measured in V
- For the three markers $M1$, $M2$ and $M3$ the values of the signals are shown in Table 6.3

Table 6.3: Signals of the Current-Based Architecture at Significant Time Steps

	M1	M2	M3
time	3.778123 μs	36.27663 μs	48.87979 μs
E_1	2.08312 μA	2.424 μA	77.6179 nA
E_2	2.14791 μA	497.56 nA	0 A
E_3	2.16392 μA	4.18272 μA	0 A
C_1	20.287 μV	3.3 V	3.3 V
T_2 C_2	19.577 μV	102.984 μV	6.69061 μV
C_3	-397.535 μV	2.71676 mV	6.75586 μV
C_1	-7.12806 μV	111.349 μV	30.3945 μV
T_1 C_2	-7.55937 μV	3.3 V	7.15152 μV
C_3	-34.1237 μV	92.249 μV	7.15323 μV
G_1	1.64174 V	1.83939 V	1.64174 V
G_2	1.64158 V	1.84247 V	1.84227 V

Table 6.4: Timing Constraints of the Current-Based Architecture

	time
Adaption phase to a new value of G_1	177.4 ns
Adaption phase to a new value of G_2	177.9 ns
Eager value drop till task discard	226.2 ns
Task discard till reallocation	177.6 ns
Total time of a reallocation process	403.8 ns
Task stealing time	157.4 ns

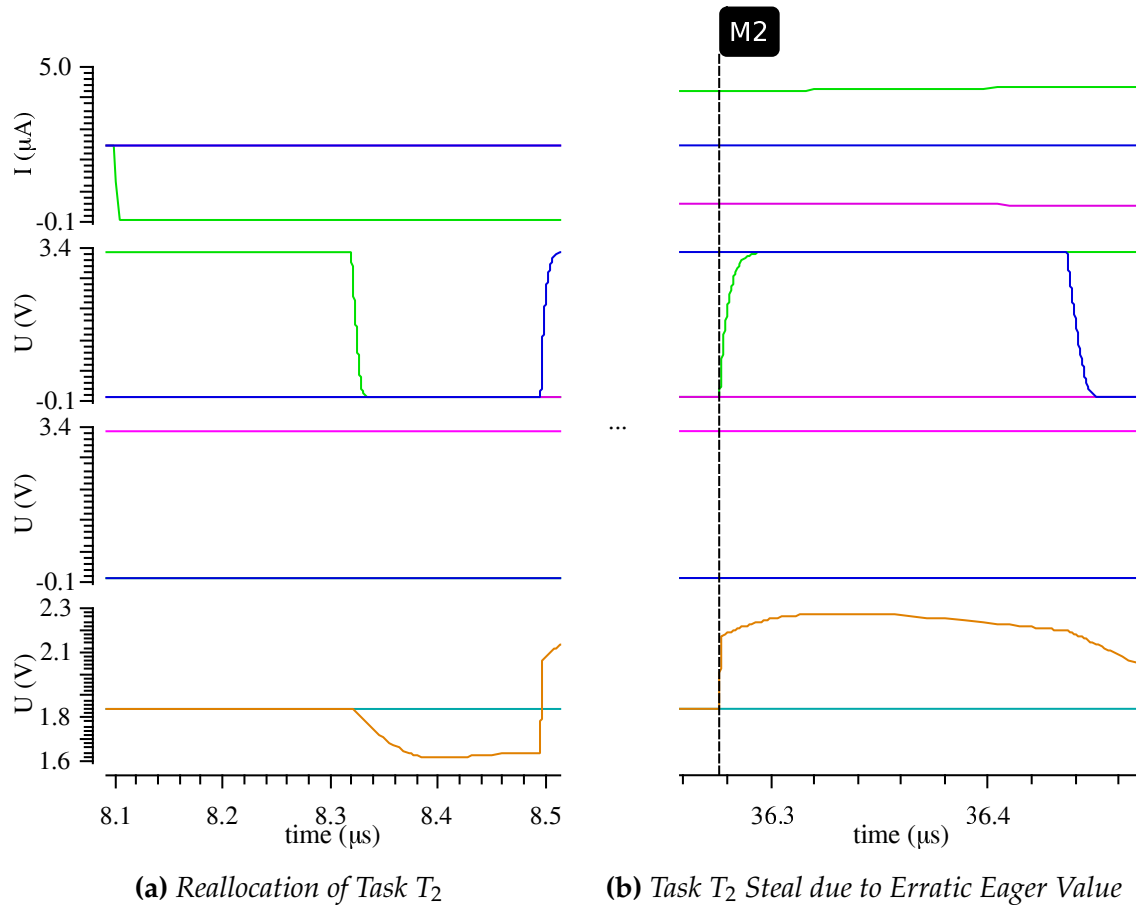


Figure 6.4: Visualized Timing Behavior of the Current-Based Architecture

- **Strip 1** displays the eager values E_1 (blue), E_2 (violet), E_3 (green) measured in μA
- **Strip 2** displays the task allocation of T_2 by C_3 (green) and C_1 (blue)
- **Strip 3** displays the task allocation of T_1 by C_2 (violet)
- **Strip 4** displays the G_i of T_1 (cyan), T_2 (orange) measured in V

Table 6.5: Signals of the Extracted View of the Voltage-Based Architecture at Significant Time Steps

	M1	M2	M3
	4.1629288 μs	33.9073 μs	46.62562 μs
E_1	1.77129 V	2.05 V	1.81622 V
E_2	1.90129 V	2.01857 V	1.70061 V
E_3	1.95629 V	2.43073 V	1.65 V
C_1	31.2757 mV	3.20849 V	30.3205 mV
T_2 C_2	89.4054 mV	2.42942 V	87.0735 mV
C_3	2.57017 V	178.34 μV	38.7243 mV
C_1	35.7576 mV	65.9322 mV	3.25879 V
T_1 C_2	387.227 mV	116.509 mV	83.2079 mV
C_3	820.422 mV	1.173 V	81.7688 mV
G_1	0.915272 V	0.94347 V	1.22839 V
G_2	1.67293 V	1.40421 V	1.6463 V

6.1.2 Simulation Runs of the Extracted View of the Architectures

Voltage-based Architecture Comparing Figure 6.1 and Figure 6.5 shows the expected impeding of the distribution behavior of the voltage-based architecture. However, a correct distribution and reallocation is still applying, if the system is operated within the feasible region as the values in Table 6.5 prove. The equivalent eager value progress as shown in Figure 6.1 is used for the extracted view simulation, with one difference: E_2 begins to rise after 1.65 μs .

Again, the allocation process is starting as the eager values exceed 1.83 V, ² but the allocation is unsuccessful until the feasible region is reached (as the value of E_3 of the marker M1 in Table 6.5 shows).

The following τ_{AAHS} timing evaluation is derived:

- $\tau_{G,1}$ requires 77.01 ns and $\tau_{G,2}$ 80.62 ns to adjust to their according new hormone levels.
- $\tau_{AAHS} = 157.63$ ns for the extracted view voltage-based architecture.

The other, changed timing constraints are given in Table 6.6.

²The core C_3 begins the allocation process for task T_1 with $E_3 = 1.83732$ V, while C_2 follows with $E_2 = 1.83476$ V.

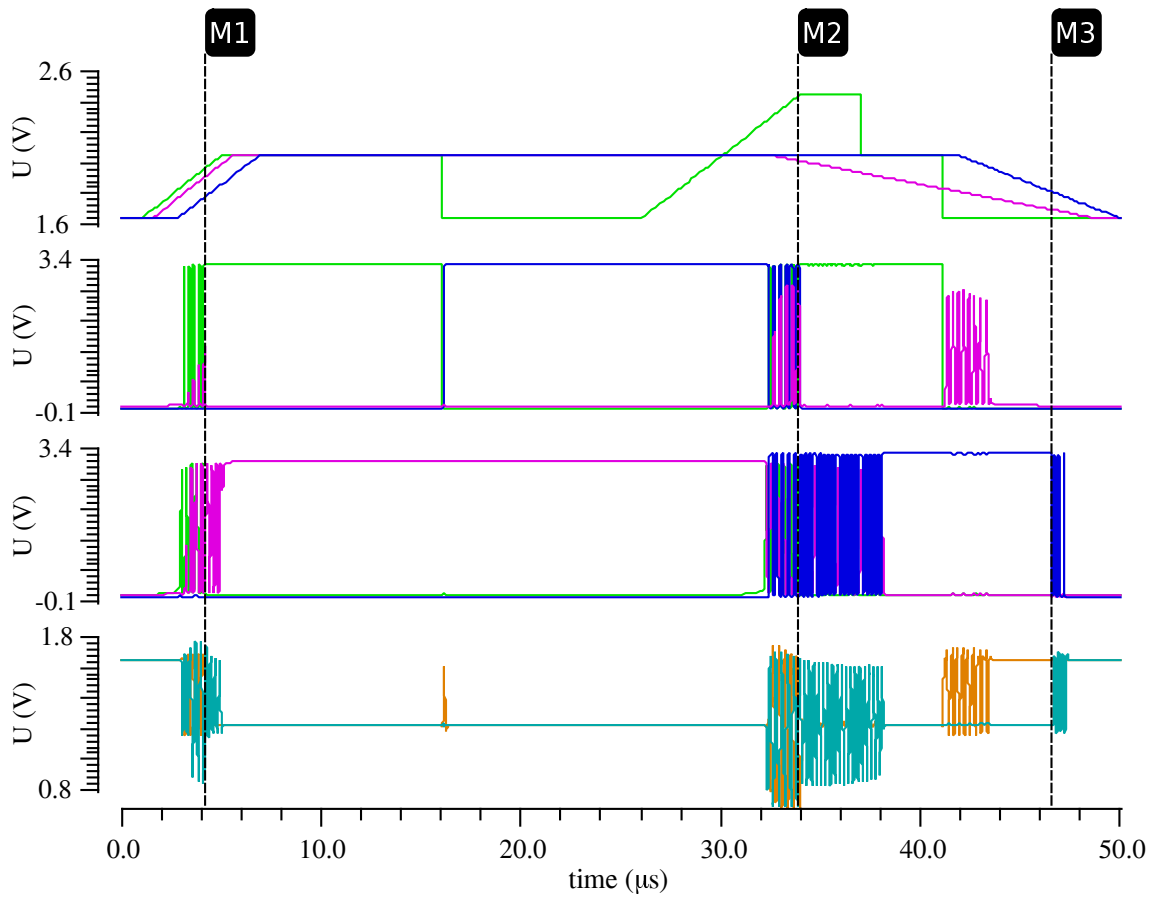


Figure 6.5: *Extracted View Simulation of the Voltage-Based Architecture*

- **Strip 1** displays the eager values E_1 (blue), E_2 (violet), E_3 (green) measured in V
- **Strip 2** displays the task allocation of T_2 by C_3 (green) and C_1 (blue)
- **Strip 3** displays the task allocation of T_1 by C_2 (violet)
- **Strip 4** displays the G_i of T_1 (cyan), T_2 (orange) measured in V
- For the markers M1, M2 and M3 the values of the signals are shown in Table 6.5

Table 6.6: *Timing Constraints of the Simulation of the Extracted View of the Voltage-Based Architecture*

	time
Eager value drop till task discard	14.55 ns
Task discard till reallocation	88.47 ns
Total time of a reallocation process	103.02 ns

Table 6.7: Hysteresis of the Schmitt Trigger of the Voltage-Based Architecture

		Schematic		Extracted View	
		$-\theta_{\gamma,i}$	$\theta_{\gamma,i}$	$-\theta_{\gamma,i}$	$\theta_{\gamma,i}$
	C_1	-	-	1.388 V	1.81 V
T_1	C_2	1.489 V	1.798 V	1.482 V	1.8 V
	C_3	1.489 V	1.803 V	1.482 V	1.798 V
	C_1	1.472 V	1.789 V	1.423 V	1.808 V
T_2	C_2	1.485 V	1.785 V	1.481 V	1.769 V
	C_3	1.478 V	1.822 V	1.482 V	1.757 V

The process of a task steal increases by approximately 20 ns to a total of 85.71 ns. Ignoring $\tau_{\text{stable},i}$ and choosing a poor inclination for the eager values shifts the distribution system into an oscillating allocation behavior for both tasks. Though, the full task steal is achieved as $E_{\text{steal},\gamma}$ increases to 2.45 V (see the values of the marker M2 in Table 6.5), but core C_3 steals task T_2 neglecting the prioritization. Further, for the other task $\tau_{\text{stable},i}$ is violated, since two cores applying with saturated eager values for the task. This oscillation is only solved by the decreasing eager value E_2 of core C_2 .

Having defined τ_{AAHS} and being able to determine SR_E with

$$\text{SR}_E = \frac{\Delta E_i}{\Delta \text{time}} = \frac{400 \text{ mV}}{200 \text{ ns}} \quad (6.10)$$

by which the eager values needs to incline,³ the simulation of schematic of the extracted view has been reissued. Figure 6.6 shows the correct allocations without any oscillating behavior and proves the importance of respecting $\tau_{\text{stable},i}$ of each task T_i and SR_E by which the eager values needs to incline (if the eager value is supposed to rise). With τ_{AAHS} defining the real-time constraints, any needed decision in less than 355.3 ns classify this architecture as not real-time capable, else the real-time bounds are hold.

Table 6.7 shows the reverse tracked hysteresis of five of the six implemented Schmitt Trigger. Only the hysteresis of C_1 for T_1 could not be determined.

Current-based Architecture Comparing both simulation runs of the current-based architecture implies no major difference and states the robustness of this

³Rising the eager value from 1.8 V to 1.95 V in approximately 75 ns leads to a gradient of 200 ns for 400 mV.

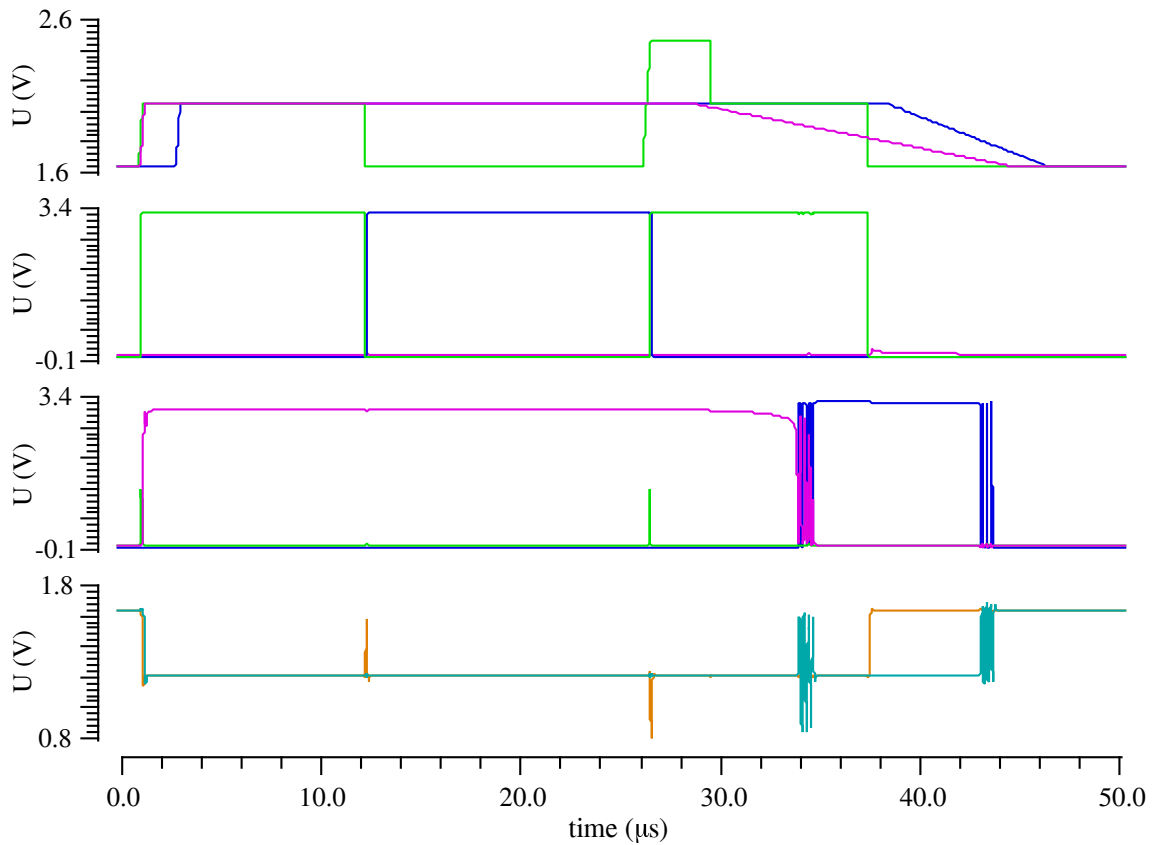


Figure 6.6: Re-Issued Simulation of the Extracted View using the Voltage-Based Architecture

- **Strip 1** displays the eager values E_1 (blue), E_2 (violet), E_3 (green) measured in V
- **Strip 2** displays the task allocation of T_2 by C_3 (green) and C_1 (blue)
- **Strip 3** displays the task allocation of T_1 by C_2 (violet)
- **Strip 4** displays the G_i of T_1 (cyan), T_2 (orange) measured in V

Table 6.8: Signals of the Extracted View of the Current-Based Architecture at Significant Time Steps

	M1	M2	M3
	3.721552 μs	36.00769 μs	49.32269 μs
E_1	1.95661 μA	2.424 μA	0 A
E_2	2.03741 μA	521.878 nA	0 A
E_3	2.11821 μA	3.96541 μA	0 A
C_1	40.1365 mV	3.28973 V	3.289997 V
T_2 C_2	38.2343 mV	36.4749 mV	37.2976 mV
C_3	40.951 mV	33.6961 mV	33.7318 mV
C_1	40.375 mV	38.3258 mV	39.6583 mV
T_1 C_2	38.3596 mV	3.28554 V	37.4701 mV
C_3	36.9561 mV	33.3989 mV	33.891 mV
G_1	1.61905 V	1.80067 V	1.62034 V
G_2	1.63413 V	1.80132 V	1.80293 V

approach. The architecture is completely insensitive, if not immune, to a poorly chosen inclination of the eager values SR_E (contrary to the voltage-based architecture). However, the costs are an altered feasibility region to be operating in with the need to heighten the eager values, as Table 6.8 shows. The eager values during the allocation, the task steal and the task discard indicate such an alteration, which is most likely caused by the allowed performance interval of the G_m of the OTAs. Assuming that 2.11757 μA is the lower bound of the feasible region and following the observations made by Table 4.1 on page 86 the optimum eager value as current hormone equals to 2.587 μA with $\beta = r_{CC} = 0.47 \mu A$ corresponding to the robustness of the architecture.

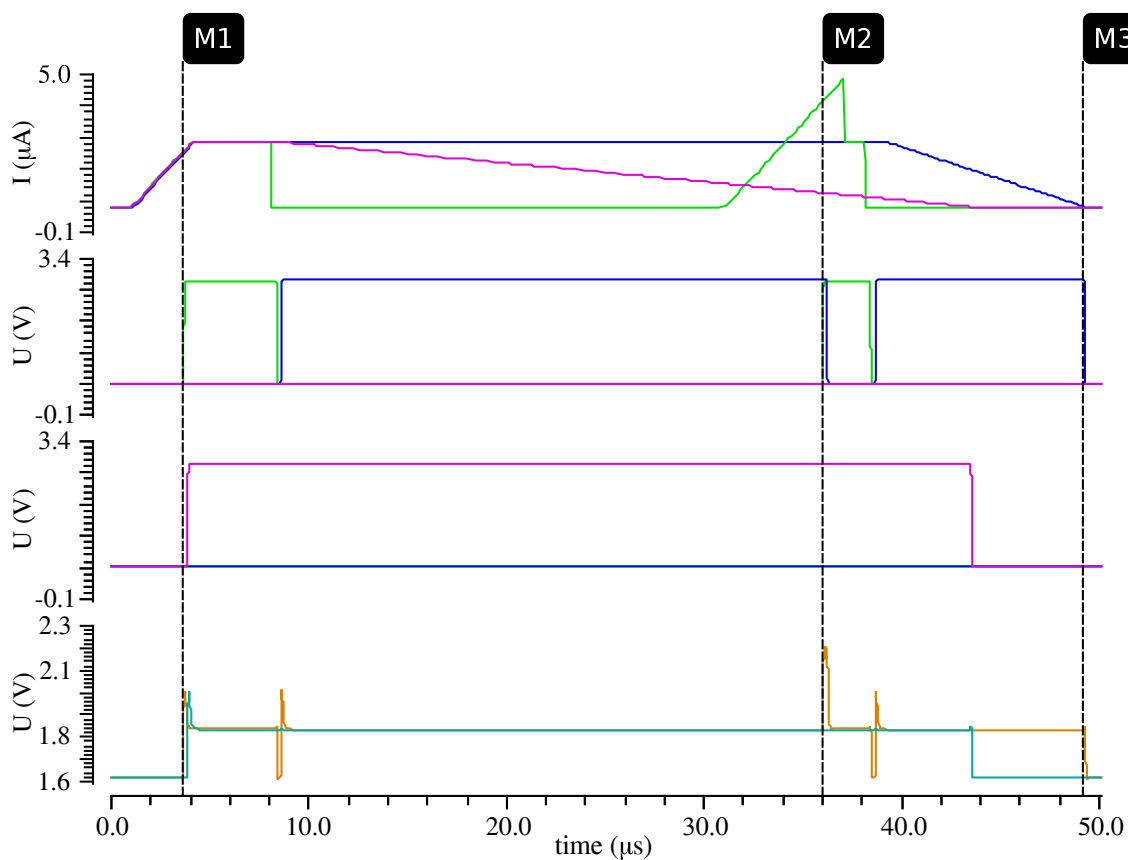
Further, minor changes are seen in concern of the timing constraints. Table 6.9 present the changed timing constraints of the simulation of the extracted view. Therefore, the timing constraints increase to $\tau_{AAHS} = 526.8 ns$.

Figure 6.7 shows the simulation run of the extracted view schematic. Again, the Global Hormone Level is displayed as the inverted voltage output of the Shunt OTA transferring the global suppressor current.

Also, three hysteresis of the Schmitt Trigger were reverse tracked, since only three decision units were allocating tasks, as presented in Table 6.10. The voltage values at the other Schmitt Trigger are also given.

Table 6.9: Timing Constraints of the Extracted View Simulation of the Current-Based Architecture

	time
Adaption phase to a new value of $\tau_{G,1}$	249.8 ns
Adaption phase to a new value of $\tau_{G,2}$	277.0 ns
Eager value drop till task discard	298.2 ns
Task discard till reallocation	217.2 ns
Total time of a reallocation process	515.4 ns
Task stealing time	204.5 ns

**Figure 6.7:** Simulation Run of the Extracted view of the Current-Based Architecture

Strip 1 displays the eager values E_1 (blue), E_2 (violet), E_3 (green) measured in μA

Strip 2 displays the task allocation of T_2 by C_3 (green) and C_1 (blue)

Strip 3 displays the task allocation of T_1 by C_2 (violet)

Strip 4 displays the G_i of T_1 (cyan), T_2 (orange) measured in V

For the three markers M1, M2 and M3 the values of the signals are shown in Table 6.8

Table 6.10: *Hysteresis of the Schmitt Trigger of the Current-Based Architecture (a) Triggering and (b) the Voltages at which the Decision Units did not trigger*

(a)

	Schematic		Extracted View	
	$-\theta_{\gamma,i}$	$\theta_{\gamma,i}$	$-\theta_{\gamma,i}$	$\theta_{\gamma,i}$
ST of C_1 for T_2	1.563 V	1.764 V	1.579 V	1.777 V
ST of C_2 for T_1	1.573 V	1.754 V	1.586 V	1.764 V
ST of C_3 for T_2	1.568 V	1.755 V	1.576 V	1.76 V

(b)

ST at C_1 of T_1	1.764 V	1.723 V	1.619 V	1.734 V
ST at C_2 of T_2	1.61 V	1.729 V	1.619 V	1.737 V
ST at C_3 of T_1	1.61 V	1.732 V	1.618 V	1.735 V

6.1.3 Hardware Measurements

Figure 6.8 shows the part of the photograph of the bounded test chip in an AMS $0.35\mu m$ analog technology, which displays the two architectures of AAHS. An extensive measurement on chip of all performances and hormone values is not possible due to several reasons:

- The closed loop operation of the OpAmps,
- The limitation of pins,
- The unknown and unexamined behavior of additional measurement pins and wires.

However, significant failing performances of components within the architectures will be noticed immediately by the failing behavior of the task distribution.

The used oscilloscope allows only to display four signals at once. Therefore, only four of the six $task_i$ on signals (see Figure 3.4) are being plotted. Also, the signals in Figure 6.9 are affected by noise. Though, the noise is not influencing the distribution mechanism in any matter and will never do, if the noise level is below the robustness value r_{CC} stated in Table 5.1 on page 107. This applies in equivalent degree to any process variation and parameter sweeps, as experienced by simulating the extracted view of the current-based architecture.

Figure 6.9 shows the reliability and the dependability of the architecture. The sub-graph on the right side shows the discard of task T_1 at core C_1 , due to a drop

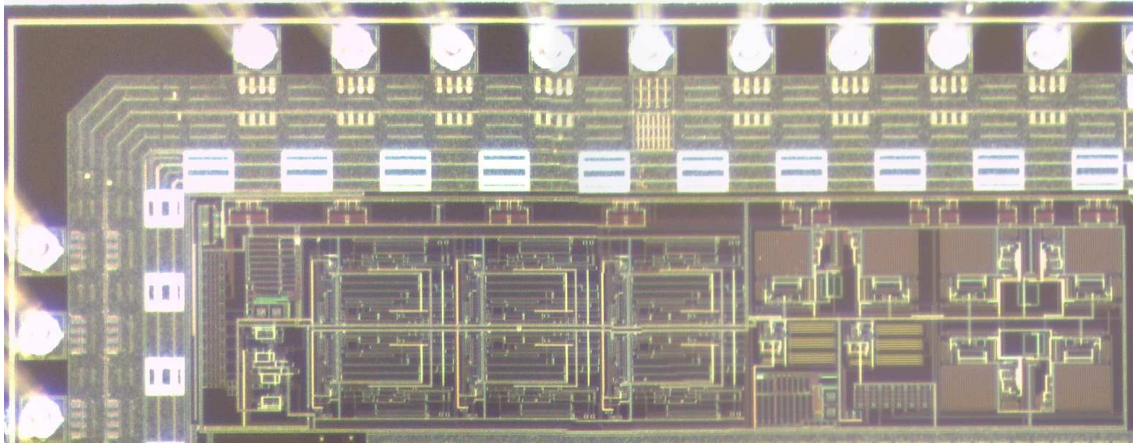


Figure 6.8: Photograph of the Test Chip of the Layout of Figure 5.8, distributing to task on three cores [vRMH15]

- the current-based architecture on the left side
- the voltage-based architecture on the right side

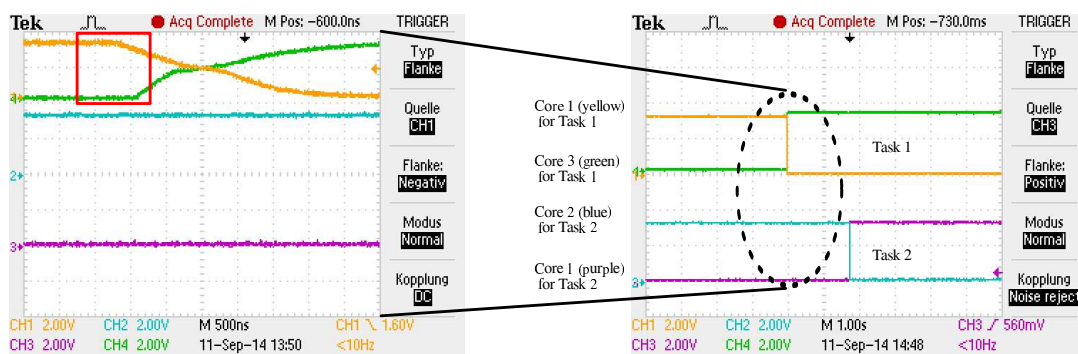


Figure 6.9: Reliability Prove of the Current-Based Architecture [vRMH15]

of the eager value E_1 (which can not be seen here). The eager and available core C_3 allocates task T_1 . Shortly afterwards, the eager value E_2 drops as well, issuing another reallocation process, since in the mean time core C_1 recovered and raised its eager value E_1 to $2.424 \mu A$ again. Any other combination of task transfers has also been measured.

A measured task reallocation is enlarged at the left side, showing the reallocation process of the two tasks between the three decision units of the cores. The solid red box at the enlarged sub-graph at the left side points out the speed of the (re-)allocation process. Noticeable less than 500 ns pass between the discard of task T_1 at core C_1 and the allocation of the task by core C_3 .

The prohibited double allocation is provable not occurring as long as the eager values are held below the upper eager value hormone bound. Otherwise a task steal by a core will appear, resulting in an immediate discard of the task at the initial core. Only, if two cores raise their eager value to twice its optimum value, which is prohibited, a double allocation occurs.

The equivalent testing is successfully conducted for the voltage-based architecture, also. Hence, for both task distribution architectures the usability is proven.

6.2 Meet the Challenge

The previous Chapter validated the reliability of AAHS. For both architectures the simulation runs and the hardware measurements proved that

- if no eager value exists, no allocation occurs (**Condition No-Allocation**),
- if only one eager value is within the feasible region, the according core is allocating successfully,
- if τ_{AAHS} is respected during the allocation processes, the allocations are done one by one, and thereby the real-time capability assured,
- both tasks are allocated by the cores (**Condition Allocating Two Tasks**), if at least two cores are applying for the tasks,
- a task steal only occurs, if the eager value exceeds the feasible region by almost twice its value (marker M2 at the Tables 6.2, 6.3, 6.5 and 6.8). Furthermore, the stealing eager value equals the value enabling double allocations, which need to be avoided (**Condition Double-Allocation**).

Further, the simulations show

- the needed eager value to allocate the tasks (marker M1 at the Tables 6.2, 6.3, 6.5 and 6.8),
- for how long a task is kept by a core with a declining eager value (marker M3 at the Tables 6.2, 6.3, 6.5 and 6.8),
- the timing constraints
 - (a) $\tau_{G,i}$ and τ_{AAHS} ,
 - (b) starting by loss of eager value to the task discard to the task allocation of another core and

Table 6.11: Comparing the Timing Constraints

	Time period of architectures	
	voltage-based	current-based
Adaption phase $\tau_{G,i}$ from G_i to η_i	80.62 ns	277 ns
Task discard till reallocation	88.47 ns	217.2 ns
real-time constraint τ_{AAHS}	157.63 ns	526.8 ns

(c) if necessary, the needed slew rate SR_E for rising the eager values.

Current Hormones versus Voltage Hormones For the timing constraints Table 6.11 states the significant differences. In terms of real-time capability (τ_{AAHS}) the voltage-based architecture is 3.3 times as fast. However, it also needs to guard the slew rate SR_E by which the eager values need to incline, contrary to the current-based architecture, which is a severe weakness.

The actual size of AAHS (\square_{AAHS}) for three cores and two tasks, build with the AMS 0.35 μm bulk CMOS technology, is:

- The voltage-based AAHS: 0.3109 mm^2 ,
- The current-based AAHS: 0.4571 mm^2 .

This prototypical layouts are quite reasonable to be used as analog building blocks on SoCs. Also, AAHS should also be redesigned and fabricated with a much smaller technology to show a comparable overhead comparison with current digital CMOS technologies [vRMH15]. In the present state however, the presented architectures are no challenge in terms of size to the assumed size of AHS published in [vRSH⁺15]. Further, the assumed sizes of AAHS architectures were not held.

The results of the percentage area calculation (equation (4.14) on page 102) of the three failure classes comparing the area of the different classified components to the total area of Chapter 4.2 are given in Table 6.12. By randomly distributed failing effects on the chip, Failure Class C effects are most likely to arise at both architectures. Only 26.14%, respectively 16.75% of the failing effects will lead to a complete loss of the voltage-based, respectively the current-based system. Concluding the fail-safety of the architectures, the current-based approach can be regarded as more reliable and higher dependability.

Table 6.12: Comparing the Failure Class Areas

Architecture	Failure Class		
	A	B	C
voltage-based	$\frac{0.0813 \text{ mm}^2}{0.3109 \text{ mm}^2} = 26.1 \%$	$\frac{0.0799 \text{ mm}^2}{0.3109 \text{ mm}^2} = 25.7 \%$	$\frac{0.0963 \text{ mm}^2}{0.3109 \text{ mm}^2} = 31 \%$
current-based	$\frac{0.0766 \text{ mm}^2}{0.4571 \text{ mm}^2} = 16.7 \%$	$\frac{0.1527 \text{ mm}^2}{0.4571 \text{ mm}^2} = 33.4 \%$	$\frac{0.1689 \text{ mm}^2}{0.4571 \text{ mm}^2} = 37 \%$

Table 6.13: Measured E_γ Lower Allocation Bounds of the Current-Based Architecture

	C_1	C_2	C_3
Test Chip 1	$E_1 = 1.8 \mu A$	$E_2 = 2.05 \mu A$	$E_3 = 2.3 \mu A$
Test Chip 2	$E_1 = 2.05 \mu A$	$E_2 = 2.05 \mu A$	$E_3 = 1.95 \mu A$
Test Chip 3	$E_1 = 1.85 \mu A$	$E_2 = 1.95 \mu A$	$E_3 = 2.0 \mu A$
Test Chip 4	$E_1 = 1.9 \mu A$	$E_2 = 2.15 \mu A$	$E_3 = 2.05 \mu A$
Test Chip 5	$E_1 = 2.2 \mu A$	$E_2 = 2.05 \mu A$	$E_3 = 2.1 \mu A$

For the current-based architecture, Table 6.13 shows the lower bounds of the eager values E_γ of several test chips at which the cores started to allocate available tasks. The change of the E_γ allocation bounds were expected, since a change was experienced during the simulation of the extracted view already. It is to be assumed that the enclosing interval of the eager values to allocate tasks and being able to react to the Global Hormone Level G_i can be derived as follows⁴:

- Current-based $E_\gamma = [2.14 \mu A, 3.06 \mu A]$

The robustness value r_{CC} equals $0.47 \mu A$ and the center point of the interval corresponds to $E_\gamma = 2.59 \mu A$.

The measured values corresponds to the values resulting from the simulation of the extracted view, presented in Table 6.8 on page 145. Yet, the measured values are affected by the inaccuracies of the measurement devices, the transmission gates and other interfering signals. Hence, the values of Table 6.13 should only be viewed as guide values.

For the voltage-based architecture, the unsuccessful allocation behavior with eager values E_γ between $[1.85 V, 1.95 V]$ has been seen during the measurements also. Though, the feasible region of $E_\gamma = 2.05 V \pm 0.1 V$ was confirmed.

⁴The lower bound of the interval is calculated by averaging the sum of the maximum of each row of Table 6.13

Table 6.14: *Advantages of the (a) voltage-based and (b) current-based Architecture*

(a)	(b)
<ul style="list-style-type: none"> • smaller in size • faster in terms of τ_{AAHS} 	<ul style="list-style-type: none"> • untouched by the inclination of the eager values SR_E • can be operated open-looped • regarded as more reliable in terms of the fail-safety of the architecture

Lastly, the measured power consumption of the test chips coincidences very closely with the simulated power consumption of the architectures. The OTAs need a total of 2.9 mA , resulting in a total power consumption of 9.57 mW , while the voltage-based architecture needs 2.1 mA resulting in 6.93 mW respectively [vRMH15].

Concluding the comparison of the two different hormone architectures it has to be stated that both approaches have their advantages. For example, contrary to previous published assumptions, the voltage-based architecture needs approximately two fifths the size of the current-based approach. Table 6.14 lists the significant advantages of each architecture. Finalizing the comparison, the current-based architecture guarantees the higher level of reliability and dependability in terms of the task distribution and fail-safety of an operating and scalable system. The trade-off for the higher level of dependability is area and speed.

Comparison of the Hormone Systems The validation of AAHS in Chapter 6.1 proves the correct functionality of the task distribution, but in terms of reliability gain any single point of failure is devastating. As the used layout shows, the global units are single point of failures for now, but at least the two-fold redundancy of the global units (Chapter 3.2.2.2 on page 68) would push the reliability gain to equal the centralized approaches.

The last missing entry of Table 3.2 is, if AAHS is considered to be scalable. Chapter 4.1.2 states the free scalability of AAHS to any number of cores N and any number of tasks $m < N$. Therefore the flexible redundancy of AHS also occurs for AAHS.

Table 6.15: Completed Summary of AAHS

	AAHS
Mechanism	decentralized symmetric
Self-control	
• Self-configuration	
• Self-optimization	hormone loops
• Self-healing	
• Self-reliance	local monitoring
Size overhead	if $O_{AAHS} < 50\%$ than $O_{AAHS} \prec O_{CB}$
Real-time bounds to assign m tasks	$\mathcal{O}(m)$
Reliability gain	depending on the redundancy factor of the global unit
Scalability	yes
Mixed-signal task migration	yes

Table 6.16: Real Comparison of AHS and AAHS

	AHS	AAHS	
		voltage-based	current-based
Needed Chip Area	0.1662 mm ²	0.3109 mm ²	0.4571 mm ²
Minimum cycle time	67 ns	80.62 ns	277.0 ns
Minimum self-configuration time	140 ns	169.09 ns	494.2 ns
Worst case task distribution time	$\mathcal{O}(m)$	$\mathcal{O}(m)$	$\mathcal{O}(m)$

The timings of AHS (minimum cycle & minimum self-configuration time) were exclusively measured by simulating the circuit model of AHS with three cores and two tasks, written in VHDL, on a FPGA.

The size overhead of AAHS equals the overhead of AHS as Table 3.1 and equation (3.9) on page 61 state. The percentage value indicates the maximum size overhead at which AAHS is still to favorable to the MTDC or AMAS.

As already stated in Chapter 2.3, the time of a single hormone cycle is defined in [vRBP11b], recited by equation (2.38). The worst case task distribution time is given with $WCTDT_{AHS} = \mathcal{O}(m)$ [BP12]. In opposite to the single hormone cycle, $\tau_{G,i}$ (equation (3.1)) defines the hormone cycle time of AAHS, while the worst case task distribution time is stated by $\sum_{i=1}^m \tau_{stable,i} = \mathcal{O}(m)$ at equation (3.5).

The minimum cycle time corresponds to the time period to reliable allocate any task and is defined by $\max_{i=1}^m (\tau_{G,i})$. Further, the minimum self-configuration time is the minimum time period to have all task safely distributed and allocated by the cores and corresponds to τ_{AAHS} also. Hence, the timings are measured in seconds and the WCTDT are notated in (\mathcal{O}) , as done in Table 6.16.

So far, the first design draft of both AAHS approaches are bigger in size and slower in configuration timings, but they are fabricated, which is still lacking for AHS, and prove the real-world functionality. A straightforward redesign in much smaller technologies would most likely favor AAHS in size and timings.

The needed chip area of AHS is appraised, oriented on Table 1 and the formulae of Chapter 4.4 in [vRSH⁺15], to match the constraints of the AAHS architectures. The minimum cycle time and self-configuration time were also presented in [vRSH⁺15, Table 1].

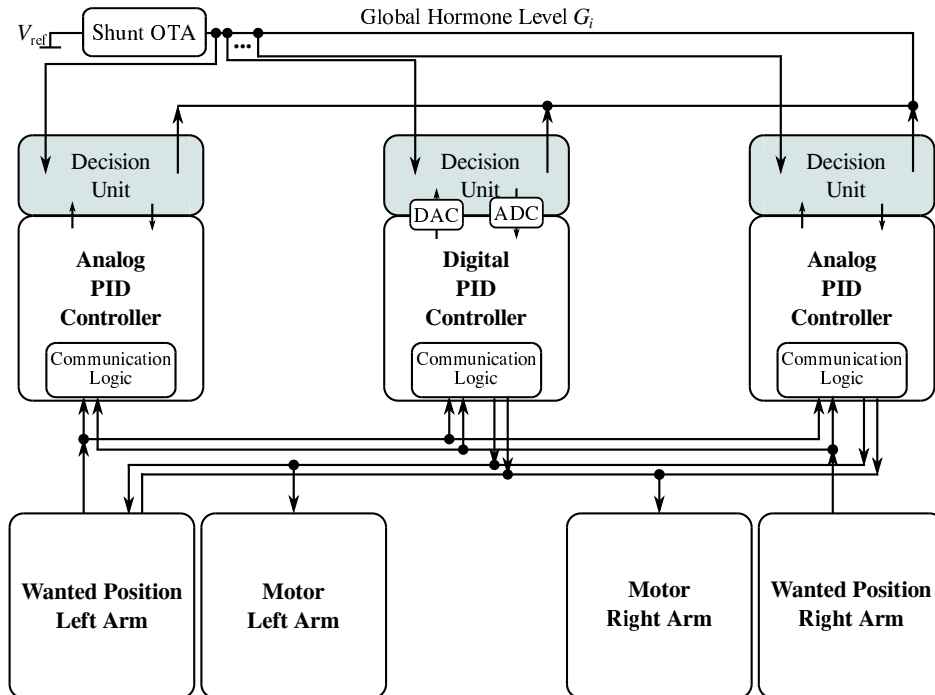


Figure 6.10: Motor Control using AAHS

6.3 Application Usage

The two examples of analog systems presented in Chapter 1.4, which are failing eventually, are now revived, yet three-folded and extended by AAHS. Figures 6.11 and 6.12 show that if an included monitor circuit catches the failing behavior and drops any eager value, a reallocation process of the task is issued. The reallocation is seen in Figure 6.11, due to a severe failing effect. The reallocation is clearly noticeable. A migration is seen in Figure 6.12(a), due to a soft failing effect. The task switch is nearly not seen at the output.

The PID Controller The motivating example of steering a gripper arm with a PID controller is extended to control two gripper arms - the left and right arm - using three PID controller. The task distribution is done by AAHS. To complicate the scenario even further, two analog and one digital PID controller are used. The architecture of the full system is seen in Figure 6.10, while Figure 6.11 shows only the simulation run of the dying core C_1 , reallocating to core C_2 , which is the digital PID controller, and the three output signals.

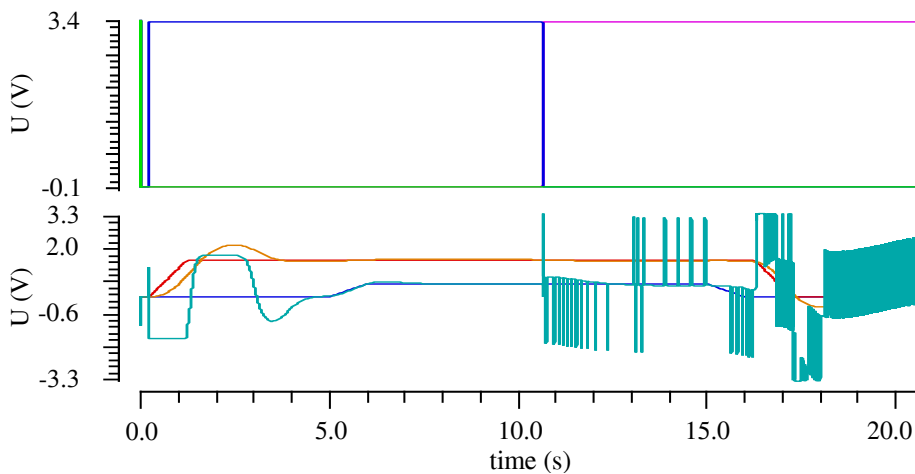


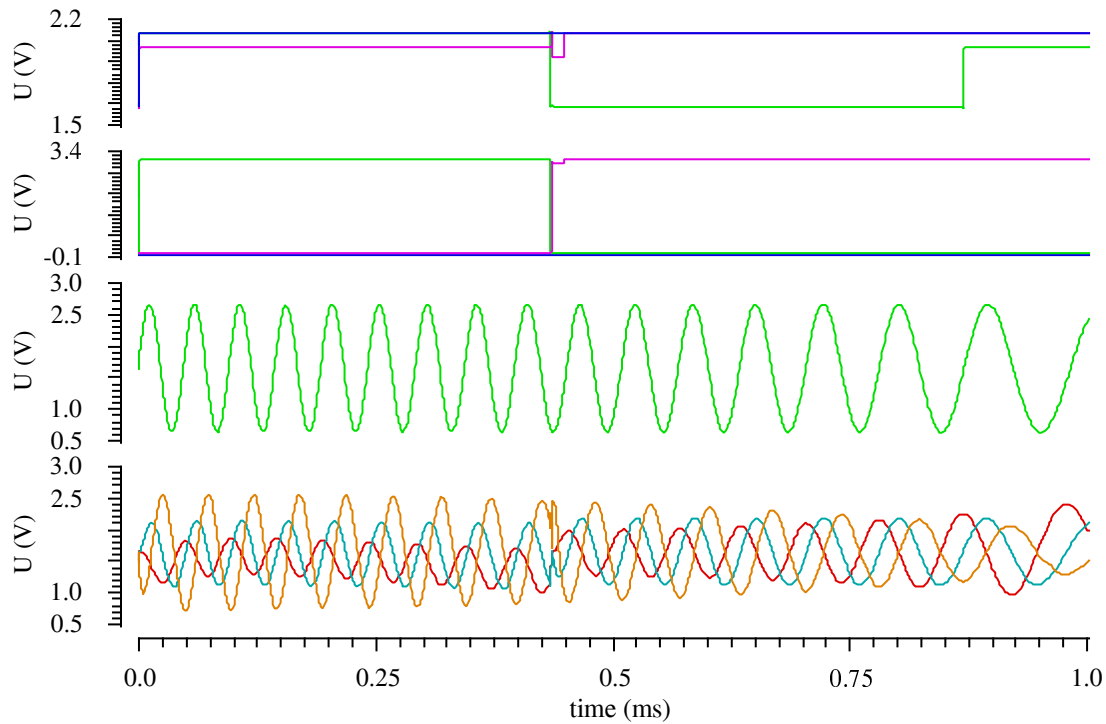
Figure 6.11: Simulation Result of the Right Arm Motor Control

- **Strip 1** displays the task allocation of the right arm by the analog core C_1 (blue), the reallocation by the digital core C_2 (violet) and the analog core C_3 (green)
- **Strip 2** displays the wanted position (red), the actual position (orange), the control voltage feeded to the motor (cyan) and an added offset torque to the motor (blue)

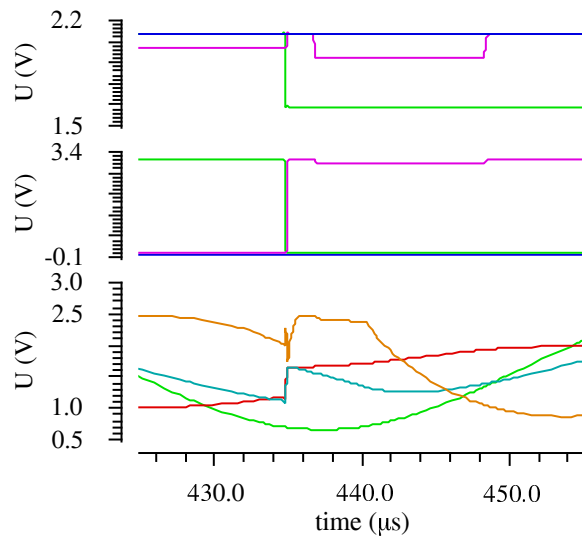
Both analog PID controller allocate the steering task for each arm. The value of the wanted position represents the degree of the angle of the arm. The values $\left(0.. \frac{V_{DD}}{2}\right)$ correspond to the angle $(0^\circ..180^\circ)$, while $\left[-\frac{V_{DD}}{2}..0\right]$ corresponds to $[180^\circ..360^\circ]$. After approximately 10.5 seconds the analog PID controller of core C_1 fails, the eager value E_1 is decreased to zero and the *right arm* task discarded. The digital PID controller of core C_2 allocates the task as soon as possible. Both motors experience an offset torque during runtime, which can nicely be seen at approximately 5 seconds lasting for 10 seconds.

The Signal Filter The second motivating example is low-pass, band-pass and high-pass filtering a frequency varying sine (Figure 6.13). Two different cut-off frequencies can be attuned. The voltage drift monitor catches the increasing offset at the differential pair of the low-pass filtering amplifier. With an offset of 100 mV the threshold voltage of a Schmitt Trigger has been reached. The monitor triggers *failing amplifier* and decreases the eager value, which results into a task reallocation.

As the task migrates at around 0.43 ms, the output signals show only little indication of the migration, as seen in Figure 6.12(b), contrary to the failing output signal seen in Figure 1.17 on page 30. Yet, the decreasing low-pass output signal is



(a)



(b)

Figure 6.12: Simulation Result of the Signal Filtering, (a) shows the complete simulation, while (b) focuses on the reallocation process

- **Strip 1** displays the eager values E_1 (blue), E_2 (violet), E_3 (green)
- **Strip 2** displays the task allocation of T_1 by C_2 (violet)
- **Strip 3 & 4** displays the input *frequency varying sine* (green) and the three output signals *high-passed* (orange), *band-passed* (cyan) and *low-passed* (red)

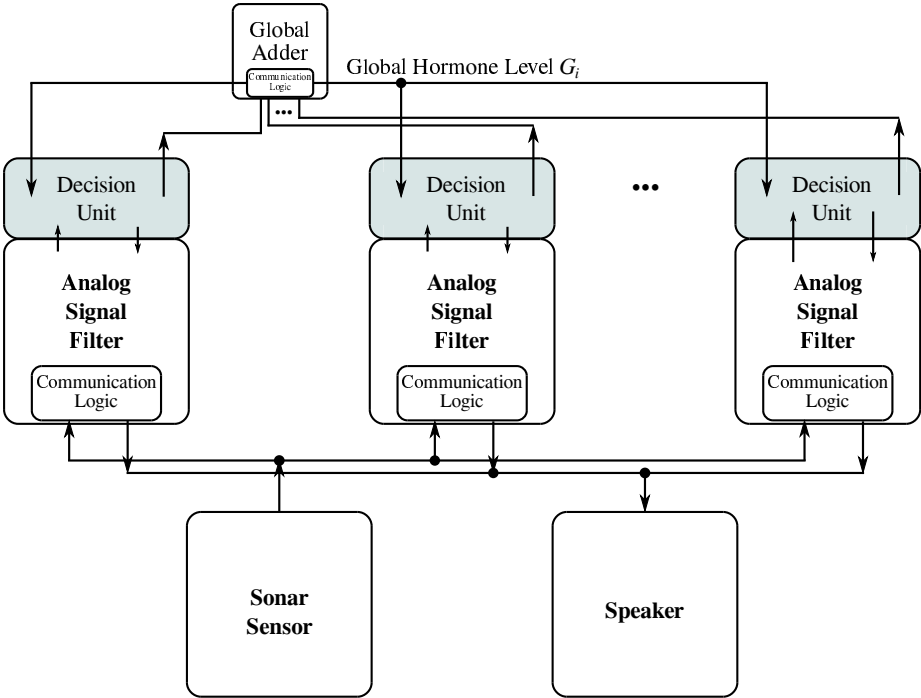


Figure 6.13: Signal Filtering using AAHS

noticeable, already indicating a failing behavior. With the migration to a working filter, the three output signals are adjusting to the correct values again.

7

Conclusions

Designing reliable architectures able to handle or avoid failures erroneous behaviors and performances is mainly done for digital systems. Hence, the goal of this thesis is a newly designed analog, reliable architecture to distribute mixed-signal tasks highly dependable within a Multi-Core SoC. Also, monitor circuits were designed to show the capability of the self-reliance, fulfilling all constraints of the self-control to be viewed as completely autonomous.

7.1 Summary

Subject of this thesis is the design of a decentral, analog architecture in regard to reliability distributing tasks highly dependable within a mixed-signal System-on-Chip. The concept of an analog, artificial hormone system continues the bio-inspired distribution of information's within a system. Any hormone represents a basic message with a particular purpose, contributed to the system. Some hormones are spread over the system for every participant to read. Others are only needed locally. Further, the hormones are used to suppress or accelerate certain behaviors, enabling stable conditions for the system to operate in. All of those in-

formation's are abstractly mapped by voltages and currents, emitted locally from each core.

The proposed architecture offers a reliable, self-controlling system in terms of autonomous operation, holding real-time bounds. Additionally, the approach is mixed-signal capable and freely scalable. To envision the stated objecties, the methodology to design the reliable, analog architecture proposed in this thesis consist of three steps:

1. The dependability analysis,
2. The design process,
3. The validation of the design.

The steps are consecutively performed, each needed to fully design the architecture from scratch all the way to fabrication, proving the feasibility of the approach and finally validating the correctness of the architecture and its functionality.

Dependability Analysis Starting with a rudimentary idealistic schematic modeling a reliable task distribution system, the design analysis allows to state the feasibility of an analog artificial hormone system. Further, a robustness value of the architecture has been defined, within which the architecture is immune to process variation, parameter sweeps and likewise effects. Following is a reliability analysis, which identified the critical components of the architecture. Failure classes have been defined, classifying the architecture and how failures affect the components and the overall system and how monitoring those components changes the classification and the dependability of the system.

Design Methodology Once, the feasibility analysis has been done, a set of parameters are derived describing the architecture. Those parameters are used to generate the specifications of every component, especially of the ten OpAmps, OTAs and STs. With all the specifications, every component has been synthesized and layouted, completing the design of the analog hormone system. Also, monitor circuits for the distribution system were implemented. The used semi-automated synthesis framework reduced the design time of the from scratch designed architectures significantly and additionally provided reliable results. Hence, the newly design OpAmps and OTAs easily allow further optimization, including the overall system, instead of the several copies of the same old, over designed textbook OpAmps and OTAs.

Validation Lastly, the design methodology has been finalized by fabricating the architecture. The measurements of the silicon, the prototype chip, are compared to the simulations of the extracted view of the hormone system, proof of feasibility of the approach and correctness of the allocation processes. Further, the timing constraints for the allocation processes, the stability constraints and the real-time capability are given. The layout also allows to estimate the actual overhead in size.

The validation reveals quite the robustness of the architecture with a manageable degree of complexity, which might help to build reliable systems insensitive to many-fold degradation and other failure sources. The reliable, analog architecture represents an excellent enhancement in optimizing the downtime, while guaranteeing the dependability. Furthermore, Table 6.15 compares the analog hormone system with the latest approaches of reliable autonomous task control architectures introduced in Chapter 1.3. Concluding this, the evaluation shows the major benefit of such a reliable architecture.

7.2 Challenges and Future Work

While this thesis presents architectures of an analog, artificial hormone system to distribute tasks highly reliable, there are some challenges to improve the reliability even further. Besides that, an implementation of an ANN as comparison approach would be interesting.

Eliminating all single points of failure So far, the implemented architectures are still suffering from single points of failure, as stated in Table 4.6 on page 93. Failing components classified as Failure Class *A* will result into an immediate fail of the distribution system and the loss of the system. Chapter 3.2.2.2 presented two approaches to eliminate the global components as single points of failure. Implementing and fabricating those would be interesting, since it allows to evaluate the differences of the four architectures in size and real-time capability with respect to the fail-safety. As lower benchmark of fail-safety, only 16.75% of occurring failing effects will strike Failure Class *A* components, which will lead to the total loss.

Enhancing the dependability analysis For the dependability analysis the parameters α and β values were standardized for all hormone loops to simplify the

design process. This also applies to the slew rates of $\tau_{G,i}$ and $\tau_{L,i}$. Enhancing the analysis to handle values of α_i and β_i enables to classify the different tasks of a system. Those classes represent the needed robustness of the execution of the tasks. The higher the needed dependability of a task, the higher the robustness class. Based on current knowledge, the values for β_i would need to be increased to maximize the robustness and minimize the effects of noise and likewise. Further, different values of $\tau_{G,i}$ and $\tau_{L,i}$ allow to classify the tasks according to the real-time capabilities and the order of importance.

Introducing global accelerators Accelerating hormones are limited to local loops only, but considering and implementing the transmission of accelerating hormones to neighboring cores makes task clustering possible as implemented in AHS. The difficulty is how to integrate them into the hormone system without affecting the reliable task distribution negatively. The formal description of the hormone loop implies the global and local hormone loops have mutual effects on one another depending on the state of the task - being free to allocate or being allocated already. The accelerators need to be weighted accordingly and attached to the Local Adder. An other idea of an accelerator could be to implement Schmitt Trigger with variable threshold voltages. Decreasing the threshold voltage corresponds to an acceleration.

Monitor circuits to enhance the self-reliance Within this thesis, a couple of monitor circuits were introduced, while one was fully integrated into the example of the signal filtering. More sophisticated monitor circuits are currently under development, which provide a more detailed health state to change the eager values E_γ accordingly in much finer steps. Also, the slew rate SR_E to surpass the fault allocation area needs to be considered, otherwise the allocation process might be pushed into an oscillating behavior, which implies the loss of the distribution and the full system.

Real-world applications and failing effects The analog hormone system has been tested with two real-world applications, only. Integrating AAHS into more real-world applications enables to state more clearly the increase of reliability and would prove the real-world usability. Further, AAHS should be attached to off chip analog cores to analyze and measure the overall reliability of the system and state the achieved performances. The results should be used to redesign AAHS focusing on the critical components. Exposing AAHS to environmental effects,

like radiation or ionization, allows to classify the reliability further. Based on current knowledge, AAHS should be able to handle a fair amount of degradation and environmental effects before failing.

Re-design the analog hormone system AAHS should be redesigned, first to eliminate the dead space within the layouts and second to decrease the size of the big components, which is, for example, the Res. OTA of the current-based architecture. As specialty, a redesign with a much smaller technology enables AAHS to be more competitive with regard to the digital implementations.

Implementing an artificial neural network The implementation on an ANN as task distribution system allows to further compare both approaches. The neural network and its fail-safety could surpass the hormone approach. So far, the sizes of both approaches are compared in regard to the possible overhead and the real-time allocation bounds, only. A real fabrication would allow a true comparison in size and speed. Maybe, both approaches could benefit from one another, if combined to be used as reliable, decentralized task distribution system.

A

Appendix

A.1 Major Functions of the Algebraic Analysis

The first function `allocateConstr` is given in Listing A.2 and defines the allocation constraints of equation (4.4) till (4.6) for all cores to each task. The forcing task drop caused by double allocations is added only for those cores, which allocated a task (see Line 11 in Listing A.2).

Next, the function `creatIneqs` defines all the inequalities needed to describe the analog hormone system. Important to notice is that the only variables left are α , β , E and θ , all others are eliminated. Further, any subscript has been eliminated as well to have the set of inequalities defined as universal as possible. Listing A.3 shoes the function `creatIneqs`.

The function `feasibleSets` solves the set of inequalities as shown in Listing A.4. Undoing all strict inequalities is needed for the latter plotting of the convex hull. However, to visualize the feasible region one of the four free variables (either θ or E) is bound to a predefined interval with fixed increments. The result, if feasible, is a set of 3D objects evolving along the fourth variable by the fixed increments, the huge set of equations and inequalities solved for each increment. The visualization of the 3D objects is done by determining the convex hull of the

Listing A.1: Complete Algebraic Analysis.

```

1 # basic constraints
2 define N, m;
3 define VDD, VSS; # upper and lower bound
4 define ε; # tolerance deviation
5 define Hγ,i := -Gi + Eγ + Aγ,i
6 define Hdγ,i := -2Gi + Eγ + Aγ,i
7
8 # loop of the different time steps
9 for t from 0 by 1 to 2 do
10   onoff[t] := define ∀cγ,i if on or off; # VDD or VSS
11   # defining the allocation constraints according equations (4.4) – (4.6)
12   bdcore[t] := allocateConstr(Hγ,i, Hdγ,i, onoff[t] at time t ∀ N, m);
13   ineqs[t] := createIneqs(bdcore[t & t-1], onoff[t] and all define);
14 end do;
15
16 # solving the inequalities
17 erg := feasibleSets(ineqs[2], ε, define fix var);
18 # the convex hull of the sets of erg
19 plot polytope3d(erg);
20
21 # loop over all feasible sets
22 for each set in erg
23   # fit the sets into the structure A x = b
24   [A,b] := convert the sets into A and b
25   # calculating the radius rCC and the coordinates of the Chebyshev center
26   [rCC, coords] := chebyCenter(A,b);
27 end do;
28
29 # generate the specifications for the OpAmps and OTAs
30 generateSpecifications(coords)

```

inequalities, showing a polyhedron of the feasible region. The amount of feasible sets depend on the fixed variable and the chosen increments for the loop. Choosing too large increments quickens the solving time, but may find no solution at all, while too small increments may overload the solution space.

Listing A.5 shows the function `chebyCenter`. The mathematical description of the Chebyshev Center is published in detail in [BTB94] and [BV04], the latter focusing on geometry and convex shapes. The Chebyshev Center is the center of the largest sphere, which is enclosed by a polytope, the largest distance to any inequality of one set. The coordinates of the center specify the values of the three free variables, while the radius r_{CC} of the Chebyshev hypersphere defines the

Listing A.2: Defining the Allocation Constraints.

```

1 #  $H_{\gamma,i} \Rightarrow$  the hormone level just before the Schmitt Trigger
2 #  $Hd_{\gamma,i} \Rightarrow$  the hormone level of a double allocation
3 #  $onoff[t] \Rightarrow$  set defining wich cores are turned on and which are turned off
4 # this function defines all the constraints for allocation tasks
5 # see (4.4) – (4.6)
6 allocateConstr := proc ( $H_{\gamma,i}$ ,  $Hd_{\gamma,i}$ ,  $onoff[t]$  at time  $t \forall N, m$ );
7
8 define  $\forall N, m$  with  $onoff[t]$ 
9 if  $C_{\gamma,i}$  is on then
10    $\theta < H_{\gamma,i}$  and
11    $Hd_{\gamma,i} < -\theta$ ;
12 else  $C_{i,j}$  is off then
13    $H_{\gamma,i} < \theta$ ;
14 fi:
15 return allocateConstr;
16 end proc:

```

robustness of the solution. The larger the value of r_{CC} , the more the three free variables may vary, while feasibility is still guaranteed. Though, two assumptions derive:

1. Since the fixed variable is substituted by the incrementing value, the algorithm to calculate the Chebyshev Center does not take the change of the fixed variable into account. The statement concerning the robustness does not apply for the fixed variable.
2. With values defined for α , β , $\theta_{\gamma,i}$, E_γ and r_{CC} the specification of the components for the synthesis (Chapter 5.1) can be generated, a very fast block level sizing process.

A.2 Determine the Number of Voters

The minimum number of voters is determined by equation (2.1):

$$N_{AV} = \begin{cases} 1 & \text{if } 3 \leq N < 9, \\ \sum_{k=1}^{\lfloor \log_3 N \rfloor} \lfloor \frac{N}{3^k} \rfloor & \text{if } N \geq 9, \end{cases} \quad (\text{A.1})$$

which is based on following assumption, wherein $3 \leq N$ is the least number of cores for voting to apply (Table 1.1). With for example nine cores each three cores

Listing A.3: Creating the Inequalities.

```

1 # bdcore[t] => the set of all the allocating constraints
2 #     => as well as of the time step before (if existing)
3 # onoff[t] => set defining wich cores are turned on and which are turned off
4 # all basic constraints defined
5 # this function defines all inequalities based on the constraints
6 createIneqs := proc(bdcore[t & t-1], onoff[t] and all define)
7
8 # defining the lower and upper bounds of  $H_{\gamma,i}$ 
9 bounds :=  $-1 < H_{\gamma,i} < 1$ ;
10 # the value of  $S_{\gamma,i}$  is either  $V_{DD}$  or  $V_{ref}$ , but can also be substituted by  $\frac{c_{\gamma,i}+1}{2}$ 
11 # substitute all  $G_i$ 
12 gl1 := subs all  $G_i$  with  $\sum_{\gamma=1}^N -\alpha \frac{c_{\gamma,i}+1}{2}$ 
13 # specify for all  $E_{\gamma}$  its value, depending on
14 # already taken a task or being free
15 gl2 := subs all  $E_{\gamma}$  with 0 or E
16 #substitute all  $A_{\gamma,i}$ 
17 gl3 := subs all  $A_{\gamma,i}$  with  $\beta c_{\gamma,i}$ 
18
19 # append the inequalities of bounds, gl1, gl2 gl3
20 ineqs := bounds U gl1 U gl2 U gl3;
21 # substitute all  $c_{\gamma,i}$  with its value in onoff[t], which is either -1 or 1
22 # now the resulting inequalities depend only on  $\alpha$ ,  $\beta$ , E and  $\theta$ 
23 ineqs := subs all  $c_{\gamma,i}$  depending on onoff[t] with either 1 or -1
24 # returning set all all inequalities
25 return ineqs;
26 end proc:

```

are connected to their own voter, while those resulting three voters are checked by one final decisive voter, totaling the number of voters to four. Those four voters will be less in size compared to one voter checking nine cores at once. The proposed mean voter is voting for voltage values only and therefore for now a special case and in general comparable to the other voters. Table A.1 shows the increase of voting units.

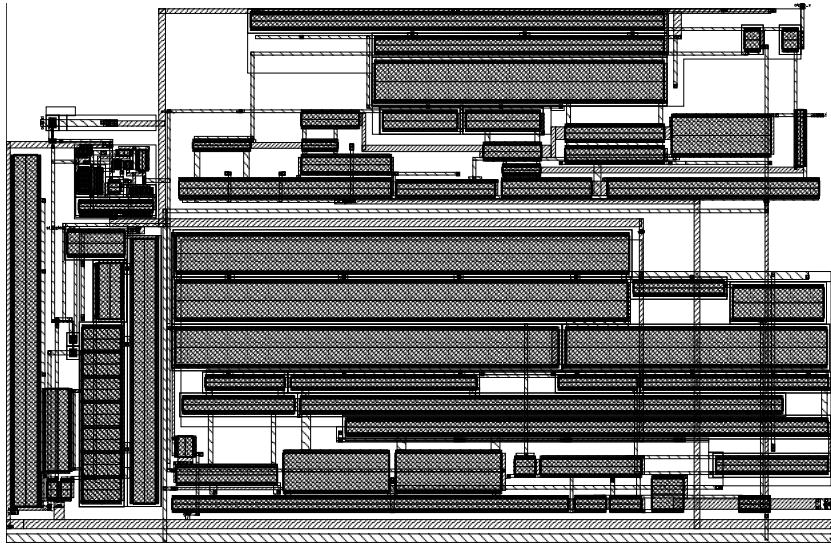
A.3 Layouts

The layout of the single decision modules are shown in more detail in Figure A.1, while Figure A.2 shows the Shunt OTA and the Global Adder.¹

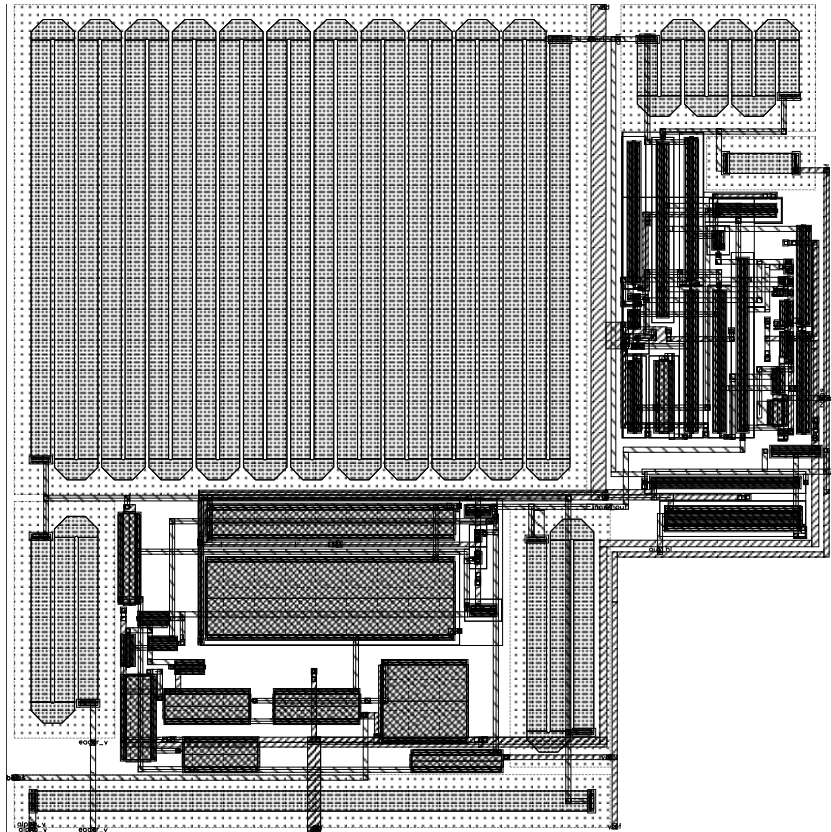
¹The size of the figures does not allow to draw conclusions about the real size of the devices.

Table A.1: Determine the Minimum Number of Voters

N	Voter				Total Number
	Level 1	Level 2	Level 3	...	
3	1				1
4	1				1
5	1				1
6	1				1
7	1				1
8	1				1
9	3	1			4
10	3	1			4
11	3	1			4
12	4	1			5
13	4	1			5
14	4	1			5
15	5	1			6
		⋮			...
25	8	1			9
26	8	1			9
27	9	3	1		13
28	9	3	1		13
29	9	3	1		13
30	10	3	1		14
		⋮			...

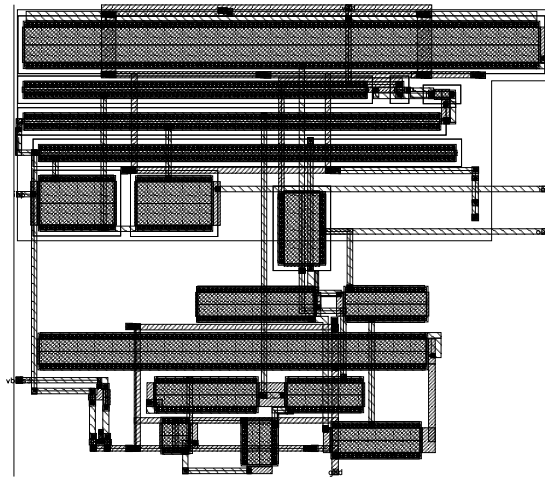


(a) *Layout of the Current-Based Decision Module*

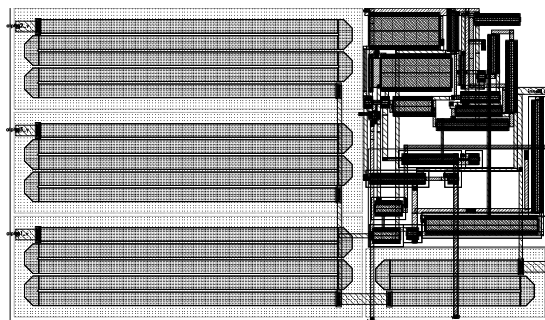


(b) *Layout of the Voltage-Based Decision Module*

Figure A.1: *Layout of the Decision Modules of the Architectures*



(a) *Layout of the Shunt OTA*



(b) *Layout of the Global Adder*

Figure A.2: *Layout of the Global Units of the Architectures*

Listing A.4: Solving and Visualizing.

```

1  # ineqs[2] => the set of all equations and inequalities
2  # ε => tolerance derivation
3  # setting the fixed variable
4  # this function solving the inequalities
5  feasibleSets := proc(ineqs[2], ε, define fix var);
6
7  ineqs2 := undo all strict inequalities using ε
8  # empty list
9  feasibleSets := [];
10 for k from 0 to 1 do
11   #substitute all fix var
12   ineqs2 := subs all fix var with k;
13   if simplex[feasible] finds solution of ineqs2 then
14     feasibleSets := [op(erg), ineqs2]
15   fi:
16 end do:
17 # returning the feasible sets
18 return feasibleSets;
19 end proc:

```

Listing A.5: Calculating the Chebyshev Center.

```

1  # matrix A => the values of the variables regarding one set
2  # vector b => the resulting values of the inequalities of one set
3  # this function determines the center of the largest hypersphere enclosed by a polyhedron
4  # based on
5  # http://www.mathworks.com/matlabcentral/fileexchange/..
6  # ..34208—uniform—distribution—over—a—convex—polytope/content/chebycenter.m
7  # seen last 2014.09.05—21:00
8  # edited to suit maple instead of matlab
9  chebyCenter := proc(A, b)
10
11  n, p := Size of Matrix A
12  # vector c corresponds to the square root of the sum of each squared cell value of each row
13  c := for each row k:  $\sqrt{\sum_{l=1}^p \text{cell}_{k,l}}$ 
14
15  A1 := Zeromatrix(1..n, 1..p+1)
16  A1(1..n, 1..p) := A
17  A1(1..n, p+1) := c
18  d := Zerovector(p+1)
19  d(p+1) := -1
20  [coords, rCC] := LPSolve(d, A1, b)
21
22 return [coords, rCC]:
23 end proc:

```

Bibliography

- [ADSN09] S. Askari, B. Dwivedi, A. Saeed, and M. Nourani. Scalable mean voting mechanism for fault tolerant analog circuits. In *Design and Test Workshop (IDT), 2009 4th International*, pages 1–6, Nov 2009.
- [ALHS12] U. Abelein, H. Lochner, D. Hahn, and S. Straube. Complexity, quality and robustness - the challenges of tomorrow's automotive electronics. In *Proc. Design, Automation and Test in Europe DATE '12*, 2012.
- [AN11] S. Askari and M. Nourani. Highly reliable analog filter design using analog voting. In *Electronics, Communications and Photonics Conference (SIEPC), 2011 Saudi International*, pages 1 – 6, april 2011.
- [Bak10] R. Jacob Baker. *CMOS Circuit Design, Layout, and Simulation*. Wiley-IEEE Press, 3rd edition, 2010.
- [BBP13] Benjamin Betting, Uwe Brinkschulte, and Mathias Pacher. Evaluation and Superiority Analysis of a Decentralized Task Control Mechanism for Dependable Real-Time SoC Architecture. In *16th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC 2013)*, Paderborn, Germany, June 2013.
- [BGL⁺06] Joseph B. Bernstein, Moshe Gurfinkel, Xiaojun Li, Jörg Walters, Yoram Shapira, and Michael Talmor. Electronic circuit reliability modeling. *Microelectronics Reliability*, 46(12):1957 – 1979, 2006.
- [BH00] I.A Basheer and M Hajmeer. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43(1):3 – 31, 2000. Neural Computing in Microbiology.

- [BP12] Uwe Brinkschulte and Mathias Pacher. An Agressive Strategy for an Artificial Hormone System to Minimize the Task Allocation Time. In *Third IEEE Workshop on Self-Organizing Real-Time Systems - SORT 2012*, Shenzhen, China, April 2012.
- [BPvR08] Uwe Brinkschulte, Mathias Pacher, and Alexander von Renteln. An Artificial Hormone System for Self-Organizing Real-Time Task Allocation in Organic Middleware. In *Organic Computing*. Springer, 2008.
- [BTB94] N. D. Botkin and V. L. Turova-Botkina. An algorithm for finding the Chebyshev center of a convex polyhedron. *Applied Mathematics and Optimization*, 29(2):211–222, March 1994.
- [BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.
- [BvRHB13] Benjamin Betting, Julius von Rosen, Lars Hedrich, and Uwe Brinkschulte. A Highly Dependable Self-Adaptive Mixed-Signal Multi-Core System-on-Chip. In *26th International Conference on Architecture of Computing Systems (ARCS 2013)*, Prague, Czech Republic, February 19th - 22nd 2013.
- [Cad] Cadence Design Framework. www.cadence.com.
- [Cau96] Gert Cauwenberghs. An analog vlsi recurrent neural network learning a continuous-time trajectory. *IEEE Transactions on Neural Networks*, 7(2):346–361, 1996.
- [CB05] Prasad Chaparala and Douglas Brisbin. Impact of nbtI and hci on pmosfet threshold voltage drift. *Microelectronics Reliability*, 45(1):13 – 18, 2005.
- [Cen03] NASA Dryden Flight Research Center. News releases: Nasa neural network project passes milestone, September 2003. Accessed: 2014-05-15.
- [CLL⁺07] Y.L. Cheng, B.L. Lin, S.Y. Lee, C.C. Chiu, and K. Wu. Cu interconnect width effect, mechanism and resolution on down-stream stress electromigration. *International Physics Symposium*, 2007.

- [CM08] Ewerson Carvalho and Fernando Moraes. Congestion-aware task mapping in heterogeneous mpsoCs. *International Symposium on System-on-Chip (SoC08)*, 2008.
- [DGLY05] Jian Deng, Desheng Gu, Xibing Li, and Zhong Qi Yue. Structural reliability analysis for implicit performance functions using artificial neural network. *Structural Safety*, 27(1):25 – 48, 2005.
- [DLS09] G. Georgakos D. Lorenz and U. Schlichtmann. Aging analysis of circuit timing considering nbti and hci. *IOLTS09*, 2009.
- [DV08] A. Das and R. Vemuri. Topology synthesis of analog circuits based on adaptively generated building blocks. *Proceedings of the 45th annual Design Automation Conference*, pages 44–49, 2008.
- [EKD⁺03] D. Ernst, Nam Sung Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge. Razor: a low-power pipeline based on circuit-level timing speculation. In *Microarchitecture, 2003. MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium on*, pages 7–18, Dec 2003.
- [EPP00] Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. Regularization networks and support vector machines. In *Advances in Computational Mathematics*, pages 1–50. MIT Press, 2000.
- [Fin01] S. Finger. *Origins of Neuroscience: A History of Explorations Into Brain Function*. Oxford University Press paperback. Oxford University Press, 2001.
- [GALH08] Stefan Grubic, Jose M. Aller, Bin Lu, and Thomas G. Habetler. A survey on testing and monitoring methods for stator insulation systems of low-voltage induction machines focusing on turn insulation problems. *IEEE Transactions on Industrial Electronics*, 55(12):4127–4136, 2008.
- [GAY89] AK. Goel and Y. T. Au-Yeung. Electromigration in the vlsi interconnect metallizations. In *Circuits and Systems, 1989., Proceedings of the 32nd Midwest Symposium on*, pages 821–824 vol.2, Aug 1989.

- [GDWL92] D. D. Gajski, N. D. Dutt, Allen C.-H. Wu, and S. Y.-L. Lin. *High-level synthesis: introduction to chip and system design*. Kluwer Academic Publishers, Norwell, MA, USA, 1992.
- [GDWM⁺08] G. Gielen, P. De Wit, E. Maricau, J. Loeckx, J. Mart'in-Mart'inez, B. Kaczer, G. Groeseneken, R. Rodr'iguez, and M. Nafr'ia. Emerging yield and reliability challenges in nanometer cmos technologies. In *DATE '08: Proceedings of the conference on Design, automation and test in Europe*, pages 1322–1327. ACM, New York, NY, USA, 2008.
- [Gie05] G.G.E. Gielen. Cad tools for embedded analogue circuits in mixed-signal integrated systems on chip. *Computers and Digital Techniques, IEE Proceedings -*, 152(3):317–332, May 2005.
- [GJP95] Federico Girosi, Michael Jones, and Tomaso Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7:219–269, 1995.
- [GK83] D. D. Gajski and R. H. Kuhn. New vlsi tools. *Computer*, 16(12):11–14, December 1983.
- [GMDW11] G. Gielen, E. Maricau, and P. De Wit. Analog circuit reliability in sub-32 nanometer cmos: Analysis and mitigation. *Proc. Design, Automation and Test in Europe DATE '11*, 2011.
- [Hay98] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1998.
- [HBKK94] B.J. Hosticka, W. Brockherde, R. Klinke, and R. Kokozinski. Design methodology for analog monolithic circuits. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, pages 387–394, 1994.
- [HG14] H. Habal and H. Gräb. MOSFET-Dimensionierung zur Verbesserung der Lebensdauer analoger Schaltungen. *Analog 2014: 14. ITG/GMM-Fachtagung Entwicklung von Analogschaltungen mit CAE-Methoden*, September 2014.

- [HL12] Ming He and Toh-Ming Lu. *Metal-dielectric interfaces in gigascale electronics: thermal and electrical stability*. Springer series in materials science. Springer, New York, NY, 2012.
- [JRSR05] N. K. Jha, P. S. Reddy, D. K. Sharma, and V. R. Rao. NBTI Degradation and Its Impact for Analog Circuit Reliability. *IEEE Transactions on Electron Devices*, 52:2609–2615, December 2005.
- [KCS] Sanjay V. Kumar, H. Kim Chris, and Sachin S. Sapatnekar. An analytical model for negative bias temperature instability. *Proceedings of the 2006 IEEE/ACM*.
- [KK11] John Keane and Chris H Kim. An odometer for cpus. *Spectrum, IEEE*, 48(5):28–33, 2011.
- [KSJ00] Eric R. Kandel, J. H. Schwartz, and Thomas M. Jessell. *Principles of Neural Science*. McGraw-Hill Medical, 4th edition, July 2000.
- [KWPK10] J. Keane, Xiaofei Wang, D. Persaud, and C.H. Kim. An all-in-one silicon odometer for separately monitoring hci, bti, and tddb. *Solid-State Circuits, IEEE Journal of*, 45(4):817–829, April 2010.
- [Lie06] Jens Lienig. Introduction to electromigration-aware physical design. In *Proceedings of the 2006 International Symposium on Physical Design, ISPD '06*, pages 39–46, New York, NY, USA, 2006. ACM.
- [LPB⁺12] Christoph Leineweber, Mathias Pacher, Benjamin Betting, Julius von Rosen, Uwe Brinkschulte, and Lars Hedrich. Detection and Defense Strategies Against Attacks on an Artificial Hormone System Running on a Mixed Signal Chip. In *International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC 2012)*, 2012.
- [Map17] Maplesoft. www.maplesoft.com, 17. Version.
- [MDJG12] E. Maricau, D. De Jonghe, and Gielen G. Hierarchical analog circuit reliability analysis using multivariate nonlinear regression and active learning sample selection. In *Proc. Design, Automation and Test in Europe DATE '11*, 2012.

- [Mea89] C. Mead. *Analog VLSI and neural systems*. VLSI systems series. Addison-Wesley, 1989.
- [MH14] M. Meissner and L. Hedrich. Feats: Framework for explorative analog topology synthesis. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, PP(99):1–1, 2014.
- [MMH11] O. Mitea, M. Meissner, and L. Hedrich. Automated Constraint-driven Topology Synthesis for Analog Circuits. In *Proc. of the Conference on Design, Automation and Test in Europe*, 2011.
- [MMLH12] M. Meissner, O. Mitea, L. Luy, and L. Hedrich. Fast isomorphism testing for a graph-based analog circuit synthesis framework. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2012*, pages 757–762, March 2012.
- [MP43] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [MS10] Janardan Misra and Indranil Saha. Artificial neural networks in hardware: A survey of two decades of progress. *Neurocomput.*, 74(1-3):239–255, December 2010.
- [Mun] MunEDA GmbH. www.muneda.com.
- [NAN08] Ali Namazi, S. Askari, and Mehrdad Nourani. Highly reliable a/d converter using analog voting. In *ICCD*, pages 334–339, 2008.
- [NC10] S.G. Narendra and A.P. Chandrakasan. *Leakage in Nanometer CMOS Technologies*. Integrated Circuits and Systems. Springer US, 2010.
- [NC13] Haiqing Nan and Kyuwon Choi. TDDDB Monitoring and Compensation Circuit Design for Deeply Scaled CMOS Technology. *Device and Materials Reliability, IEEE Transactions on*, 13(1):18–25, March 2013.
- [OM03] T.R. Oldham and F.B. McLean. Total ionizing dose effects in mos oxides and devices. *Nuclear Science, IEEE Transactions on*, 50(3):483–499, June 2003.

- [Par91] B. Parhami. Voting networks. *Reliability, IEEE Transactions on*, 40(3):380–394, Aug 1991.
- [Par92] B. Parhami. Optimal algorithms for exact, inexact, and approval voting. In *Fault-Tolerant Computing, 1992. FTCS-22. Digest of Papers., Twenty-Second International Symposium on*, pages 404–411, July 1992.
- [Phe06] Mc Pherson. Reliability challenges for 45nm and beyond. *Design Automation Conference (DAC)*, 2006.
- [Pom93] Dean Pomerleau. Knowledge-based training of artificial neural networks for autonomous robot driving. In J. Connell and S. Mahadevan, editors, *Robot Learning*. 1993.
- [PWMC07] Antonis Papanikolaou, Hua Wang, Miguel Miranda, and Francky Catthoor. Reliability issues in deep deep sub-micron technologies: time-dependent variability and its impact on embedded system design. In *IOLTS '07: Proceedings of the 13th IEEE International On-Line Testing Symposium*, page 121, Washington, DC, USA, 2007. IEEE Computer Society.
- [QS08] Zhenyu Qi and Mircea R. Stan. Nbti resilient circuits using adaptive body biasing. In *GLSVLSI '08: Proceedings of the 18th ACM Great Lakes symposium on VLSI*, pages 285–290, New York, NY, USA, 2008. ACM.
- [RCN04] Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolic. *Digital integrated circuits- A design perspective*. Prentice Hall, 2ed edition, 2004.
- [Roj96] Raúl Rojas. *Neural Networks: A Systematic Introduction*. Springer-Verlag New York, Inc., New York, NY, USA, 1996.
- [RSS06] P.S. Rajpal, K.S. Shishodia, and G.S. Sekhon. An artificial neural network for modeling reliability, availability and maintainability of a repairable system. *Reliability Engineering & System Safety*, 91(7):809 – 819, 2006.

- [Rut06] R. A. Rutenbar. Design automation for analog: the next generation of tool challenges. *ICCAD '06 Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, page 460, 2006.
- [SBCD04] G. Stone, E. A. Boulter, I. Culbert, and H. Dhirani. *Electrical Insulation for Rotating Machines: Design, Evaluation, Aging, Testing, and Repair*. IEEE Press Series on Power Engineering. Wiley, 2004.
- [SdBF04] I Skliarova and A de Brito Ferrari. Reconfigurable hardware sat solvers: a survey of systems. *Computers, IEEE Transactions on*, 53(11):1449–1461, Nov 2004.
- [SGRG10] Christian Schluender, Wolfgang Gustin, Hans Reisinger, and Tibor Grasser. A new physics-based Nbti model for dc- and ac-stress enabling accurate circuit aging simulations considering recovery. *Proc. Zuverlässigkeit und Entwurf (ITG-FB 231)*, pages 33–40, 2010.
- [SH11] Felix Salfelder and Lars Hedrich. An NBTI model for efficient transient simulation of analogue circuits. In *Proc. edaWorkshop 11*, pages 27 – 32. VDE Verlag, 2011.
- [Shu13] Timmy Shumate. Simulationsbasierter Entwurf eines Demonstrators zur hormongesteuerten, dezentralen Verteilung und zuverlässigen Bearbeitung von Regelungsaufgabenn. Diplomarbeit, Institut für Informatik, Lehrstuhl Entwurfsmethodik, Goethe Universität Frankfurt, September 2013.
- [SJL08] K. Stawiasz, K.A Jenkins, and Pong-Fei Lu. On-chip circuit for monitoring frequency degradation due to nbti. In *Reliability Physics Symposium, 2008. IRPS 2008. IEEE International*, pages 532–535, April 2008.
- [SLM99] A. Schmid, Y. Leblebici, and D. Mlynek. Mixed analogue-digital artificial-neural-network architecture with on-chip learning. *Circuits, Devices and Systems, IEEE Proceedings -*, 146(6):345–349, Dec 1999.
- [SM88] J.R. Srour and James M. McGarrity. Radiation effects on microelectronics in space. *Proceedings of the IEEE*, 76(11):1443–1469, Nov 1988.

- [Smi97] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, San Diego, CA, USA, 1997.
- [SMN11] Rodrigo Martins da Silva, Luiza de Macedo Mourelle, and Nadia Nedjah. Compact yet efficient hardware architecture for multilayer-perceptron neural networks. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, 22:647 – 663, Dec 2011.
- [SS01] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.
- [Ste11] S. Steinhorst. *Formal verification methodologies for nonlinear analog circuits*. Doktorarbeit, Institut für Informatik, Lehrstuhl Entwurfsmethodik, Goethe Universität Frankfurt, 2011.
- [SvRH14] T. Shumate, J. von Rosen, and L. Hedrich. A Highly Dependable Reactive Architecture Layer for Autonomous Robots based on an Artificial Analog Hormone System. *Analog 2014: 14. ITG/GMM-Fachtagung Entwicklung von Analogschaltungen mit CAE-Methoden*, September 2014.
- [VCMS⁺99] A Vittal, L.H. Chen, M. Marek-Sadowska, Kai-Ping Wang, and S. Yang. Crosstalk in vlsi interconnections. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 18(12):1817–1824, Dec 1999.
- [vdM90] Christoph von der Malsburg. Network self-organization. In S. F. Zornetzer, J. Davis, and C. Lau, editors, *An Introduction to Neural and Electronic Networks*, pages 421–432. Academic Press, 1990.
- [VMS97] A Vittal and M. Marek-Sadowska. Crosstalk reduction for vlsi. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 16(3):290–298, March 1997.
- [vR12] Alexander von Renteln. *Eine organische Middleware für verteilte Echtzeitsysteme*. Doktorarbeit, Institut für Informatik, Lehrstuhl Eingebettete Systeme, Goethe Universität Frankfurt, Januar 2012.

- [vRB07] Alexander von Renteln and Uwe Brinkschulte. Reliability of an Artificial Hormone System with Self-X Properties. In *Parallel and Distributed Computing and Systems*, Cambridge, Massachusetts, USA, November 19 - 21 2007.
- [vRBBH12] Julius von Rosen, Benjamin Betting, Uwe Brinkschulte, and Lars Hedrich. Ein hochverlässliches, selbst-adaptives, Mixed-Signal Mehrkern-System-on-Chip. In 6. *GMM/GI/ITG-Fachtagung für Zuverlässigkeit und Entwurf (ZuE 2012)*, Bremen, Deutschland, September 2012.
- [vRBP11a] Alexander von Renteln, Uwe Brinkschulte, and Mathias Pacher. Introducing a Simplified Implementation of the AHS Organic Middleware. In *International Conference on Autonomic Computing (ICAC), Workshop on Organic Computing (OC)*, Karlsruhe, June 2011.
- [vRBP11b] Alexander von Renteln, Uwe Brinkschulte, and Mathias Pacher. The Artificial Hormone System - An Organic Middleware for Self-organising Real-Time Task Allocation. *Organic Computing - A Paradigm Shift for Complex Systems*, Springer Verlag, Basel, 2011.
- [vRH12] Julius von Rosen and Lars Hedrich. A Highly Dependable, Analog Artificial Hormone System as Middleware for a Multi-Core System-on-Chip. In *IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC 2012)*, Santa Cruz, USA, October 2012. PhD Forum.
- [vRMH15] J. von Rosen, M. Meissner, and L. Hedrich. Semiautomatic implementation of a bioinspired reliable analog task distribution architecture for multiple analog cores. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2015*, Grenoble, France, March 9 - 13 2015.
- [vRSH⁺15] Julius von Rosen, Felix Salfelder, Lars Hedrich, Benjamin Betting, and Uwe Brinkschulte. A highly dependable self-adaptive mixed-signal multi-core system-on-chip architecture. *Integration, the {VLSI} Journal*, 48(0):55 – 71, 2015.
- [WH06] Xiaoying Wang and L. Hedrich. An approach to topology synthesis of analog circuits using hierarchical blocks and symbolic analysis.

In *Proc. Asia and South Pacific Conference on Design Automation*, page 6pp., 24–27 Jan. 2006.

- [YFB⁺09] Baoguang Yan, Qingguo Fan, J.B. Bernstein, Jin Qin, and Jun Dai. Reliability simulation and circuit-failure analysis in analog and mixed-signal applications. *Device and Materials Reliability, IEEE Transactions on*, 9(3):339–347, Sept. 2009.
- [ZDL90] S.F. Zornetzer, J.L. Davis, and C. Lau. *An Introduction to neural and electronic networks*. Academic Press, 1990.

Lebenslauf

von Rosen, Julius

geboren am 16. August 1984 in Frankfurt am Main

Ausbildung

- 2010 - 2014 Promotionsstudium im Fachgebiet Informatik an der Goethe-Universität in Frankfurt am Main.
- 2005 - 2010 Studium der Informatik mit Nebenfach Betriebswirtschaftslehre an der Goethe Universität in Frankfurt am Main. Abschluss als Master of Science im Mai 2010. Masterarbeit an der Professur für Entwurfsmethodik: „Spezifikationsgesteuerte Abstraktionsverfeinerung für die formale Verifikation analoger Schaltungen“. (Betreuer: Dipl.-Inf. Sebastian Steinhorst)
- 2002 - 2005 Studium der Betriebswirtschaftslehre an der Goethe Universität in Frankfurt am Main.
- 2000 - 2002 Besuch des Rosseau Lake College, Ontario, Kanada, Abschluss Ontario Secondary School Diploma (Allgemeine Hochschulreife).
- 1994 - 2000 Besuch des Gymnasialen Zweiges des Lessing Gymnasiums Frankfurt am Main.

Berufliche Tätigkeiten

- seit Nov. 2010 Wissenschaftlicher Mitarbeiter an der Professur für Entwurfsmethodik (Prof. Dr.-Ing. Lars Hedrich) an der Goethe-Universität in Frankfurt am Main.