

Inhaltsverzeichnis

Vorwort	i
Kurzfassung	iii
Abstract	v
1 Elektronikentwicklung und Trends in der Automobilindustrie	1
1.1 Anforderungen an zukünftige E/E-Fahrzeugarchitekturen	1
1.2 Herausforderung Multicore-Technologie in Domänensteuergeräten	6
1.3 Aufbau und Struktur der Arbeit	13
1.4 Wissenschaftliche Beiträge der Arbeit	13
2 Stand der Technik	17
2.1 Redundanz und Zuverlässigkeit sicherheitsrelevanter Systemen	17
2.1.1 Majoritätsredundanz	20
2.1.2 Safety-Mechanismen gängiger Multicore-Architekturen	23
2.1.3 Grundbegriffe technischer Zuverlässigkeit	26
2.1.4 Funktionale Sicherheit in der Automobilindustrie	27
2.2 Virtualisierungsmethoden im Überblick	29
2.2.1 Grundbegriffe Virtualisierung	29
2.2.2 Vollvirtualisierung	33
2.2.3 Binärübersetzung	33
2.2.4 Paravirtualisierung	35
2.2.5 Microkernel	36
2.2.6 Virtualisierungsunterstützung in Hardwarearchitekturen	37
2.2.7 Ansätze in der Automobilelektronik	41
2.3 Parallelisierung von Software	54
2.3.1 Funktionsmodellierung und Softwareentwicklung für Steuergeräte	56
2.3.2 Software-Parallelisierung im Fahrzeug	57
2.4 Multicore-Controller für sicherheitsrelevante Domänensteuergeräte	59
2.4.1 Infineon AURIX TC27x	60
2.4.2 NXP/Freescale MPC57xx Calypso	63

3 Virtualisierung und Parallelisierung für Fail-Operational Multicore-DCUs	65
3.1 Architekturanforderungen für Multicore-DCUs	65
3.2 Das 2oo2DFS-System als Fail-Operational-Ansatz in der Fahrwerkselektronik	74
3.2.1 Realisierung einer ECU-internen 2oo2DFS-Architektur für EPS-Steuergeräte	75
3.2.2 Systemzuverlässigkeit des ECU-internen 2oo2-Ansatzes . .	76
3.3 Hypervisor-basierte Virtualisierung in sicherheitsrelevanten Fahrwerksteuergeräten	91
3.3.1 Pre-Virtualisierung als Alternative zur Paravirtualisierung .	91
3.3.2 I/O-Virtualisierung in Domänensteuergeräten	100
3.4 Software-Parallelisierung bei hochintegrierten Fahrwerksteuergeräten	110
3.4.1 Integration	112
3.4.2 Implementierung mit ereignisgesteuerten Tasks	114
3.4.3 Laufzeitmessung mit Stoppuhren	115
4 Experimentelle Validierung und Ergebnisse	117
4.1 Auswertung des Redundanzkonzepts für Fail-Operational	117
4.2 Ergebnisse der Pre-Virtualisierung	121
4.3 Auswertung der Hypervisor-basierten I/O Virtualisierung	126
4.3.1 Messergebnisse der Implementierung	126
4.3.2 Auswertung der Sendelatzenzen des Timing-Modells	130
4.3.3 Vergleich der Messwerte mit der Timing-Analyse	136
4.4 Auswertung der Software-Migration und Parallelisierung	138
5 Zusammenfassung und Ausblick	143
5.1 Zusammenfassung Fail-Operational	143
5.2 Zusammenfassung Virtualisierungskonzepte	144
5.3 Zusammenfassung Software-Parallelisierung	146
5.4 Ausblick	147
A Fragebogen und Ergebnisdiagramme	151
B Berechnung und Validierung der Zustandswahrscheinlichkeiten	171
C Messergebnisse	181
D Ergebnisse der Timing-Analyse	183
E Vergleichstabellen	205
Abbildungsverzeichnis	212

Inhaltsverzeichnis	ix
Tabellenverzeichnis	214
Abkürzungsverzeichnis	215
Literaturverzeichnis	219
Eigene Veröffentlichungen	227