

Inhaltsverzeichnis

1. Allgemeine Grundlagen	11
1.1. Echtzeit	11
1.2. Phasengetriebenes Prozessmodell: das V-Modell	11
1.3. Buildprozess: vom Modell zum Executable	13
1.4. Zusammenfassung	24
2. Mikroprozessortechnik Grundlagen	26
2.1. Aufbau von Mikroprozessoren	26
2.2. Codebearbeitung	29
2.3. (Speicher-) Adressierung, Adressierungsart	31
2.4. Wait-states, Burstzugriffe	37
2.5. Cache	38
2.6. Pipeline	43
2.7. Interrupts	45
2.8. Traps/Exceptions	46
2.9. Datenkonsistenz	46
2.10. Gegenüberstellung Desktopprozessoren - Embedded Prozessoren	48
2.11. Zusammenfassung	50
3. Betriebssysteme	51
3.1. Kein OS: Endlosschleife plus Interrupts	51
3.2. OSEK/VDX	54
3.3. Kooperatives und präemptives Multitasking	60
3.4. POSIX	68
3.5. Zusammenfassung	74
4. Timingtheorie	75
4.1. Timingparameter	75
4.2. Stochastische Aspekte	82
4.3. CPU-Last	87
4.4. Buslast	96
4.5. Logical Execution Time (LET)	97
4.6. Zusammenfassung	98
5. Timinganalysetechniken	100
5.1. Übersicht, Einteilung in Ebenen	100
5.2. Begriffsklärungen	104
5.3. Statische Codeanalyse	106
5.4. Codesimulation	116
5.5. Laufzeitmessung	130
5.6. Hardware basiertes Tracing	143
5.7. Tracing basierend auf Instrumentierung der Software	164
5.8. Schedulingssimulation	183
5.9. Statische Schedulinganalyse	192
5.10. Evolutionäre Algorithmen zur Optimierung	206

5.11. Anordnung der Timinganalysetechniken im V-Modell	208
6. Praxisbeispiele Timingprobleme	211
6.1. Woher kommen nur die ganzen Interrupts?	211
6.2. OSEK ECC: selten die beste Wahl	212
6.3. Seltene Abstürze 17 Minuten nach Reset	215
6.4. Fehlende bzw. doppelt empfangene Sensordaten	217
6.5. Im Rennen mit angezogener Handbremse	223
6.6. Messung liefert größere WCET als statische Codeanalyse	224
6.7. Netzwerkmanagementbotschaften kommen manchmal zu früh	226
6.8. Lückenloser Timingprozess in einem Serienprojekt	227
6.9. Timinganalyse spart Automobilhersteller 12 Mio. Euro	228
6.10. Zusammenfassung	229
7. Timing bei Multicore, Manycore, Multi-ECU	230
7.1. Multicore Grundlagen	230
7.2. Verschiedene Arten paralleler Ausführung	236
7.3. Datenkonsistenz, Spinlocks	245
7.4. Klonen von Speicheradressen	250
7.5. Zusammenfassung	254
8. Laufzeitoptimierung	255
8.1. Laufzeitoptimierung auf der Schedulingebene	255
8.2. Laufzeitoptimierte Speichernutzung	260
8.3. Laufzeitoptimierung auf der Codeebene	266
8.4. Zusammenfassung und „Fahrplan“ für die Laufzeitoptimierung	287
9. Methodik im Entwicklungsprozess	290
9.1. Anforderungen mit Timingbezug	290
9.2. Zusammenarbeit von Projektpartnern	300
9.3. Timingkonzept, Scheduling Layout, OS Konfiguration	301
9.4. Laufzeitdebugging	302
9.5. Laufzeitoptimierung	303
9.6. Laufzeitabsicherung	303
9.7. Frühzeitige Berücksichtigung zukünftiger Funktionalität	305
9.8. Timingabsicherung in der Serie	307
9.9. Positivbeispiel: CoReMa von Vitesco Technologies	309
9.10. Zusammenfassung	311
10. AUTOSAR	312
10.1. AUTOSAR Classical Platform (CP)	313
10.2. AUTOSAR Adaptive Platform (AP)	316
10.3. TIMEX (AUTOSAR Timing Extensions)	326
10.4. ARTI (AUTOSAR/ASAM Run-Time Interface)	328
10.5. Technical Report „Timing Analysis“	335
10.6. Zusammenfassung	335

11. Safety, ISO 26262	337
11.1. Grundlagen	337
11.2. Sicherheitsstandards und Timing – Timingabsicherung	339
11.3. Werkzeuge für die Timingabsicherung	342
11.4. Rechtliche Aspekte	342
11.5. Zusammenfassung	343
12. Ausblick	344
A. Glossar	347