

# Contents

<b>1</b>	<b>基础知识</b>	<b>10</b>
1.1	实时系统	10
1.2	阶段性的软件开发模型: V-Model	10
1.3	编译过程: 从模型到可执行文件	12
1.4	总结	23
<b>2</b>	<b>处理器基础知识</b>	<b>24</b>
2.1	处理器的构造	24
2.2	代码执行	26
2.3	存储器寻址及其模式	29
2.4	等待状态, 突发访问	35
2.5	缓存	35
2.6	流水线 (Pipeline)	41
2.7	中断	42
2.8	陷阱/异常 (Exception/trap)	43
2.9	数据一致性	44
2.10	与桌面处理器相比 - 嵌入式处理器	46
2.11	总结	47
<b>3</b>	<b>操作系统</b>	<b>48</b>
3.1	无操作系统: 无限循环加中断	48
3.2	OSEK/VDX	51
3.3	多任务: 协作与抢占	57
3.4	POSIX	65
3.5	总结	71
<b>4</b>	<b>软件时间理论</b>	<b>72</b>
4.1	时间参数	72
4.2	统计参数	78
4.3	CPU 负载	82
4.4	总线负载	89
4.5	逻辑执行时间	89
4.6	总结	91
<b>5</b>	<b>软件时间分析方法</b>	<b>92</b>
5.1	概览及在不同层面上的分类	92
5.2	术语定义	95
5.3	静态代码分析	97
5.4	代码仿真	106

5.5	运行时间测量	114
5.6	基于硬件的追踪	126
5.7	基于软件方法的追踪	140
5.8	调度模拟	156
5.9	静态调度分析	164
5.10	使用进化算法进行优化	174
5.11	时间分析方法在 V 模型中的应用	177
<b>6</b>	<b>软件时间问题案例</b>	<b>179</b>
6.1	中断都是哪来的?	179
6.2	OSEK ECC: 并非最优选择	180
6.3	重置 17 分钟后发生偶发崩溃	183
6.4	遗漏及重复的传感器数据	184
6.5	拉着手刹去比赛	189
6.6	实际测量得到的 WCET 比静态代码分析得到的更大	190
6.7	有时候网络管理报文来的太早了	192
6.8	系列项目中无间断的时间分析	192
6.9	时间分析使得车厂节省了 1200 万欧元	193
6.10	总结	194
<b>7</b>	<b>多核及多 ECU 环境下的软件时间</b>	<b>195</b>
7.1	多核基础知识	195
7.2	并发执行的多种类型	200
7.3	数据一致性, Spinlocks	208
7.4	存储地址克隆	214
7.5	总结	216
<b>8</b>	<b>软件运行时间优化</b>	<b>218</b>
8.1	调度层级的运行时间优化	218
8.2	针对存储器使用的运行时间优化	222
8.3	代码层级的运行时间优化	227
8.4	运行时间优化总结与指南	246
<b>9</b>	<b>开发过程中的方法技巧</b>	<b>249</b>
9.1	与时间相关的各种需求	249
9.2	开发期间的协作	257
9.3	软件运行时间概念、调度设计和操作系统配置	258
9.4	软件运行时间调试	259
9.5	运行时间优化	259
9.6	时间验证	260
9.7	尽早考虑后期的功能	262

9.8	最终产品中的时间监测 . . . . .	263
9.9	一个好例子: Vitesco Technologies 出品的CoReMa . . . . .	265
9.10	总结 . . . . .	266
<b>10</b>	<b>AUTOSAR</b>	<b>268</b>
10.1	AUTOSAR CP . . . . .	269
10.2	AUTOSAR AP . . . . .	270
10.3	AUTOSAR 时间扩展 TIMEX . . . . .	280
10.4	AUTOSAR/ASAM 运行时间接口 ARTI . . . . .	282
10.5	技术报告“时间分析” . . . . .	285
10.6	总结 . . . . .	286
<b>11</b>	<b>功能安全, ISO 26262</b>	<b>287</b>
11.1	基础知识 . . . . .	287
11.2	安全标准、时间及时间验证 . . . . .	289
11.3	时间安全所需工具 . . . . .	291
11.4	法律方面的考量 . . . . .	292
11.5	总结 . . . . .	292
<b>12</b>	<b>前景</b>	<b>293</b>