

Inhalt

1	Einleitung	1
1.1	Behandelte Fragestellungen	1
1.1.1	Spezifikation	1
1.1.2	Entwurfsverfahren	2
1.1.3	Maschinelle Synthese	4
1.1.4	„Backend“	8
1.1.5	Wiederverwendung eines Modells	9
1.2	Ergänzende Literatur	12
1.3	Entwurf	13
1.3.1	Modellierung	16
2	Rolle eines Modells bei der Verifikation	21
2.1	Verifikationsmethoden	22
2.1.1	„Review“	23
2.1.2	Versuch	25
2.1.3	Beweis	28
2.2	Getrennte Verifikation von Funktion, Timing und Aufwand	32
2.3	Entwurfsfehler, Stimuli und Waveforms	35
2.3.1	Arten von Entwurfsfehlern	36
2.3.2	Verfahren der Stimulierzeugung	37
2.3.3	Erleichterte Inspektion der Signale durch Komprimierung	51
2.4	Simulation, ASIC-Emulation oder Prototyp?	58
3	Modelle: Verbergen und Vernachlässigen	73
3.1	Signale	74
3.2	Strukturmodelle und deren Konfiguration	83
3.3	Verhaltensmodelle: Abstraktion statt Hierarchie	98
3.4	Simulatorkonzepte	102
3.5	Signalfußrichtung	108
3.6	Designprozesse	116
3.7	Abstrakte Modellierung	120
3.7.1	Arten der Abstraktion	121
4	Strukturinformation (SI): Geometrie bis Kombinatorik	123
4.1	Abstraktionsmechanismen	124
4.2	Geometrie	127
4.3	Topologie	129
4.4	Gatterebene	130
4.4.1	Modellierung der Verzögerungszeit	131
4.4.2	Verschiedene Gattermodelle	133
4.4.3	Abstraktionsgrad der Gatter-„ebene“	136
4.5	Kombinatorik und Register (SI.CR)	140
4.5.1	Kombinatorik	141
4.5.2	Simulations- versus Synthesesemantik	142
4.5.3	Einfache und komplexe Kombinatorik	148

4.5.4	Vor- und Nachteile der Abstraktion	149
4.5.5	Register	156
5	Strukturinformation (SI): FSM und Erweiterte FSM	159
5.1	Synchroner Entwurf	161
5.1.1	Transienter und stabiler Zustand	162
5.1.2	Abstraktions-„ebene“: RT-Ebene	164
5.2	„Multi Process“-Modellierung (SI.MP)	167
5.2.1	Kombinatorik: Unvollständige „sensitivity list“	169
5.2.2	Risiko: Akkumulation von Timingpfaden	172
5.3	„Single-Process“-Modellierung (SI.SP)	176
5.3.1	Vollständige Spezifikation eines Ausgangswerts	182
5.3.2	Gemeinsamer Teilausdruck („common subexpressions“)	188
5.3.3	Speicherung von Variablen in einem getakteten Prozeß	190
5.3.4	Ergänzte Moore-FSM: Entwurfssicherheit und schnelle Reaktion	199
5.4	EFSM: Getrennter Kontroll- und Datenzustand	202
5.4.1	EFSM: Explizite Datenpfade (SI.ED)	206
5.4.2	EFSM: Implizite einfache Datenpfade (SI.ISD)	207
6	(SI.ICS) Implizite Modellierung des Kontrollzustands: Herleitung	213
6.1	Kompakte Modellierung und Rescheduling	216
6.2	„Wait“-Anweisung statt „sensitivity list“	219
6.3	„single-process“-Schablone einer (E)FSM	223
6.4	Modellierung mit bedingten Sprüngen	226
6.5	Modellierung mit strukturierten Sprachmitteln	229
6.6	Modellierung ohne eine lineare Zustandssequenz	231
7	Analyse der Kontrollpfade	237
7.1	Analogie auf der Gatterebene: Timinganalyse	237
7.2	Kontrollpfad-Analyse	240
7.2.1	Gemeinsamer Teilausdruck und Mehrfachnutzung	242
7.2.2	Untersuchung der Realisierbarkeit	248
7.3	Zyklen in vollständigen Kontrollpfaden	254
7.3.1	„Statisch bestimmte“ Ausdrücke	255
7.3.2	Schleifen ohne „wait“-Anweisung	257
7.3.3	Schleifen mit „wait“-Anweisungen	259
7.4	Notwendiges Scheduling und Durchsatzanpassung	260
7.4.1	Algorithmus, Durchsatz und Effizienz	262
7.4.2	Empirische Pfadverteilungen und die Taktfrequenz optimaler Effizienz	267
7.4.3	Durchsatzanpassung durch Neu-Synthese?	272
8	Umformung durch Rescheduling	277
8.1	Analogie auf der Gatterebene: Retiming	279
8.2	Timing bei (SI.ICS)	280
8.3	Regeln des Re-scheduling	283
8.3.1	Funktionales und zeitliches Klemmenverhalten	284
8.3.2	Lineare Sequenz von Anweisungen	285
8.3.3	Sequenz von Anweisungen mit einer bedingten Verzweigung	290
8.4	Durchsatzanpassung durch Rescheduling	295

8.4.1	Verringerung des Durchsatzes	296
8.4.2	Durchsatzreduzierung am Beispiel des PID-Reglermoduls	300
8.4.3	Erhöhung des Durchsatzes	314
8.4.4	Schnittstellenüberbuchung und Datenabhängigkeiten	317
8.5	Automatisches Rescheduling	327
9	(SI/ICS): Schleifen, Beispiel und Initialisierung	337
9.1	„while“- und „for“-Schleifen	338
9.2	Implizite Zustandsmodellierung an einem Beispiel	345
9.2.1	Modellierung der Einheit „transmitter“	345
9.2.2	Kontrollpfad-Analyse	347
9.3	SI/ICS: Vor- und Nachteile	352
9.4	„reset“- und „interrupt“-Modellierung	355
9.5	Bemerkungen	365
9.5.1	Graphisches versus textuelles Verhaltensmodell	366
10	SI/ICD: Abstraktion „DSP-Block“?	371
10.1	Implementierung eines Algorithmus	371
10.2	Emulation einer „Datenfluß“-Simulation	378
10.3	Implizite Modellierung komplexer Datenpfade	383
10.4	EFSM-Konzept und Wert/Zeit-Relation	388
10.5	Abstraktionsgrad verschiedener Blöcke	394
10.6	Abstraktionsstufen der Strukturinformation (SI)	397
10.7	Bemerkungen	399
10.7.1	Protokollierung und graphische Nachbearbeitung	399
11	Vereinfachte Modellierung eines funktionalen Aspekts (FC)	403
11.1	Verhalten nach der Aktivierung der „reset“-Leitung	404
11.2	Systemebene: Programmierbare Einheit	408
12	Abstrakte Datentypen (AD)	419
12.1	Definition durch Bitfelder	419
12.2	Abstrakter Datentyp	422
12.3	Datentypen als ein Abstraktionsmechanismus	424
12.3.1	Übergang zu konkreten Datentypen an einem Beispiel	428
12.4	Übersicht: Stufen und Übergänge	435
12.5	Polymorphe Signale	436
13	Modellierung der Wert/Zeit-Relation (VT)	443
13.1	Software versus Hardware	444
13.2	Zeitskalen der simulierten Zeit	446
13.2.1	Reduktion der Ereigniszahl	446
13.3	Zweidimensionale Zeitskala in VHDL	448
13.3.1	Das „gepufferte“ Taktsignal	453
13.4	Abstraktionstufen der Wert/Zeit-Relation	455
13.5	Transformationen	460
13.6	Stufen der Zerlegung („Decomposition“)	462
13.6.1	Einführung eines Taktes	469

14 Modellierung mit dem 'U'-Wert	475
14.1 Ursachen und Vermeidung zufälligen Verhaltens	476
14.2 'U'-Wert auf verschiedenen Abstraktionsebenen	481
14.3 Ausreichende Initialisierung	483
14.4 Erweiterung des Wertebereichs	484
14.4.1 IEEE Standardwertesatz: „std_ulogic“	484
14.4.2 Abstrakte Datentypen	487
14.4.3 Simulations- versus Synthesesemantik	488
14.5 Erweiterung der Modelle	491
14.6 RT-, Gatterebene und die Realität	493
14.6.1 Einfluß einer logischen Umformung auf das 'U'-Verhalten	495
14.6.2 Aussagekraft einer Simulation mit 'U'	499
14.7 Einführung von „std_ulogic“-Typen	502
14.7.1 Mehrfachnutzung von Registern und Datenpfaden	505
14.8 Abstrakte Datentypen oder „std_ulogic“-Typen?	510
14.9 Bemerkungen	516
15 Entwurfsstudie: „WM“	519
15.1 Schrittweises Zerlegen und Verfeinern	519
15.2 Spezifikation	522
15.3 VT.F: Früher Prototyp	527
15.3.1 Einfache Implementation des WM	527
15.3.2 Speicherverwaltung mit „access types“	530
15.3.3 „pseudo-random“ Testbench	534
15.4 VT.C: Ausgliederung des Objektes „WM“	539
15.5 VT.FMA: Schnittstellen und Allokation	547
15.5.1 Getrennte Schnittstellen	548
15.5.2 Polymorpher Datentyp „t_wmEntry“	558
15.5.3 Funktion des Algorithmus „first free list“	565
15.6 VT.MI: Zerlegung der Einheit WM und Scheduling	572
15.6.1 Synchrone Verpackung von RAMs	574
15.6.2 Zugriff auf ein verpacktes RAM	580
15.6.3 Einplanung und „arbitration“ im WMC	583
15.7 Diskussion der Entwurfsstudie WM	592
15.7.1 Implementationstechniken Hard- versus Software	595
15.7.2 Hardware/Software Co-Design	598
Literaturverzeichnis	603
Sachverzeichnis	609