

《Verilog编程艺术》

图书基本信息

书名：《Verilog编程艺术》

13位ISBN编号：978712122061X

出版时间：2014-1

作者：魏家明

页数：456

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《Verilog编程艺术》

内容概要

本书深入地探讨了Verilog编程，分为七个部分：设计原则、语言特性、书写文档、高级设计、时钟和复位、验证之路、其他介绍。本书对这些部分做了重点的探讨：Verilog编码风格、Verilog-2001的新特性、简洁高效的编程、容易出错的语言元素、可配置设计、时钟生成、复位设计、验证方法等。另外，本书还对SystemVerilog做了简单的介绍。

书籍目录

目 录

第一部分 设计原则

第1章 美的设计

2

1.1 美学观点

2

1.2 美是修养

3

1.3 专业术语

4

第2章 高效之道

5

2.1 敏捷开发

5

2.2 代码质量

6

2.3 版本控制

7

2.4 提早集成

7

第3章 组织管理

9

3.1 植物分类

9

3.2 SoC特性

11

3.3 设计流程

12

3.4 仔细规划

12

3.5 管理表格

13

3.6 模块层次

14

3.7 目录组织

14

第4章 使用工具

19

4.1 使用Emacs

19

4.1.1 Emacs介绍

19

4.1.2 Emacs安装

19

4.1.3 常用快捷键

20

4.1.4 我的.emacs

21	
4.1.5	cua-base.el
22	
4.1.6	verilog-mode.el
23	
4.1.7	shell buffer
23	
4.2	使用Shell
24	
4.2.1	Shell介绍
24	
4.2.2	Shell例子
24	
4.2.3	Perl例子
25	
4.3	使用CVS
26	
4.3.1	CVS介绍
26	
4.3.2	CVS术语
27	
4.3.3	CVS初始化
27	
4.3.4	CVS常用命令
29	
	第5章 编码风格
31	
5.1	干干净净
32	
5.2	代码划分
32	
5.3	代码要求
33	
5.3.1	Verilog部分
33	
5.3.2	SystemVerilog部分
40	
5.4	名字定义
40	
5.5	书写格式
42	
5.5.1	模块端口声名
42	
5.5.2	模块实例化
45	
5.5.3	函数和任务调用
47	
5.5.4	书写语句
47	

5.5.5 书写表达式

48

5.6 添加注释

49

5.7 参数化

50

5.8 lint检查

52

第二部分 语言特性

第6章 Verilog特性

54

6.1 Verilog标准

54

6.2 抽象级别

54

6.3 可综合子集

55

6.4 保持一致

57

第7章 常数

58

7.1 整数 (integer)

58

7.2 实数 (real)

60

7.3 字符串 (string)

60

7.4 标识符 (identifier)

60

第8章 数据类型

61

8.1 线网 (net)

61

8.1.1 wire和tri

61

8.1.2 wor、wand、trior、triand

61

8.1.3 tri0、tri1

61

8.1.4 uwire

61

8.1.5 supply0、supply1

62

8.1.6 驱动强度

62

8.1.7 默认net

62

8.2 变量 (variable)

62

8.3 线网和变量的区别	63
8.4 向量 (vector)	64
8.5 数组 (array)	65
8.6 多维数组	65
第9章 表达式	67
9.1 操作符 (Operator)	67
9.1.1 操作符的优先级 (Operator priority)	68
9.1.2 表达式中使用整数	68
9.1.3 算术操作符 (Arithmetic operators)	69
9.1.4 算术表达式中的regs和integers	69
9.1.5 比较操作符 (Compare operators)	70
9.1.6 逻辑操作符 (Logical operators)	70
9.1.7 位运算操作符 (Bitwise operators)	71
9.1.8 归约操作符 (Reduction operators)	71
9.1.9 移位操作符 (Shift operators)	71
9.1.10 条件操作符 (Conditional operator)	72
9.1.11 连接操作符 (Concatenations)	72
9.2 操作数 (Operands)	73
9.2.1 向量的抽取 (bit-select and part-select)	73
9.2.2 part-select的例子	75
9.2.3 数组的访问	75
9.2.4 字符串	76
9.3 表达式位长 (Expression bit lengths)	77
9.3.1 表达式位长规则	77
9.3.2 表达式位长问题的例子A	

78	
9.3.3	表达式位长问题的例子B
79	
9.3.4	表达式位长问题的例子C
79	
9.3.5	表达式位长问题的例子D
79	
9.3.6	表达式位长问题的例子E
80	
9.4	符号表达式 (Signed expressions)
80	
9.4.1	表达式类型规则
81	
9.4.2	计算表达式的步骤
81	
9.4.3	执行赋值的步骤
82	
9.4.4	signed表达式中处理x和z
82	
9.4.5	signed应用的例子
82	
9.4.6	signed应用的错误
83	
9.5	赋值和截断 (Assignments and truncation)
84	
9.6	与x/z比较
85	
第10章	赋值操作
86	
10.1	连续赋值
86	
10.2	过程赋值
87	
第11章	门级和开关级模型
88	
11.1	门和开关的声明语法
88	
11.1.1	门和开关类型
88	
11.1.2	驱动强度
88	
11.1.3	延迟
89	
11.1.4	实例数组
89	
11.2	and、nand、nor、or、xor、xnor
90	
11.3	buf、not
90	

11.4 bufif1、bufif0、notif1、notif0	90
11.5 MOS switches	90
11.6 Bidirectional pass switches	91
11.7 pullup、pulldown	91
第12章 用户定义原语	92
12.1 UDP定义	92
12.1.1 UDP状态表	92
12.1.2 状态表符号	93
12.2 组合UDP	93
12.3 电平敏感时序UDP	93
12.4 沿敏感时序UDP	94
第13章 行为模型	97
13.1 概览	97
13.2 过程赋值	98
13.2.1 阻塞赋值	98
13.2.2 非阻塞赋值	99
13.3 过程连续赋值	102
13.3.1 assign和deassign过程语句	103
13.3.2 force和release过程语句	103
13.4 条件语句	104
13.5 循环语句	105
13.5.1 for循环例子	106
13.5.2 disable语句	107
13.6 过程时序控制	108
13.6.1 延迟控制 (Delay control)	

108	
13.6.2	事件控制 (Event control)
108	
13.6.3	命名事件 (Named events)
109	
13.6.4	事件or操作符 (Event or operator)
109	
13.6.5	隐含事件列表 (Implicit event_expression list)
109	
13.6.6	电平敏感事件控制 (Level-sensitive event control)
111	
13.6.7	赋值间时序控制 (Intra-assignment timing controls)
111	
13.7	块语句
113	
13.7.1	顺序块 (Sequential block)
113	
13.7.2	并行块 (Parallel block)
114	
13.7.3	块名字 (Block names)
114	
13.7.4	开始和结束时间 (Start and finish times)
114	
13.8	结构化过程
116	
13.8.1	initial construct
116	
13.8.2	always construct
116	
13.8.3	always的敏感列表
117	
13.8.4	并发进程
117	
13.9	always有关的问题
118	
13.9.1	敏感列表不完整
118	
13.9.2	赋值顺序错误
119	
第14章	case语句
120	
14.1	case语句定义
121	
14.2	case语句的执行
122	
14.3	Verilog和VHDL对比
123	
14.4	case的应用
123	

14.5 casez的应用	125
14.6 描述状态机	126
14.7 casex的误用	127
14.8 casez的误用	128
14.9 full_case 和parallel_case	128
14.10 full_case	129
14.10.1 不是full的case语句	129
14.10.2 是full的case语句	129
14.10.3 使用full_case综合指令	130
14.10.4 full_case综合指令的缺点	131
14.10.5 使用full_case指令后还是生成Latch	132
14.11 parallel_case	132
14.11.1 不是parallel的case语句	132
14.11.2 是parallel的case语句	133
14.11.3 使用parallel_case综合指令	133
14.11.4 parallel_case综合指令的缺点	134
14.11.5 没有必要的parallel_case指令	135
14.12 综合时的警告	135
14.13 case语句的编码原则	136
第15章 task和function	137
15.1 task和function之间的不同点	137
15.2 task的声明和使能	137
15.2.1 task的声明	137
15.2.2 task的使能和参数传递	138
15.2.3 task的内存使用和并发进程	

140	
15.3	disable语句
141	
15.3.1	disable语句的例子A
141	
15.3.2	disable语句的例子B
143	
15.4	function的声明和调用
145	
15.4.1	function的声明
145	
15.4.2	function的返回值
147	
15.4.3	function的调用
147	
15.4.4	function的规则
147	
15.4.5	constant function
148	
15.5	task的误用
149	
15.6	function的误用
149	
第16章	调度和赋值
151	
16.1	仿真过程
151	
16.2	事件仿真
151	
16.3	仿真参考模型
152	
16.4	分层事件队列
153	
16.4.1	事件队列分类
154	
16.4.2	事件队列特性
155	
16.4.3	事件调度例子
155	
16.5	确定性和不确定性
157	
16.5.1	确定性 (Determinism)
157	
16.5.2	不确定性 (Nondeterminism)
157	
16.6	赋值的调度含义
158	
16.6.1	连续赋值
159	

16.6.2 过程连续赋值	159
16.6.3 阻塞赋值	159
16.6.4 非阻塞赋值	159
16.6.5 开关处理	159
16.6.6 端口连接	159
16.6.7 任务和函数	160
16.7 阻塞赋值和非阻塞赋值	160
16.7.1 阻塞赋值	160
16.7.2 非阻塞赋值	161
16.8 赋值使用原则	161
16.9 自己触发自己	162
16.10 仿真零延迟RTL模型	163
16.11 惯性延迟和传输延迟	165
16.11.1 门级仿真中的传输延迟	166
16.11.2 各种#delay的位置	168
16.11.3 仿真时钟生成方法	169
16.12 延迟线模型	170
16.13 使用#1延迟	171
16.14 多个公共时钟和竞争条件	172
16.15 避免混杂阻塞赋值和非阻塞赋值	173
16.16 RTL和门级混合仿真	176
16.16.1 RTL-to-Gates仿真	177
16.16.2 Gates-to-RTL仿真	177
16.16.3 有时钟偏差的门级时钟树	178
16.16.4 有时钟偏差的Vendor模型	

178	
16.16.5	错误的Vendor模型
179	
16.16.6	结论和建议
183	
16.17	带有SDF延迟的门级仿真
183	
16.17.1	全系统仿真
183	
16.17.2	软件要花钱
184	
16.17.3	门级回归仿真
184	
16.18	验证平台技巧
185	
16.18.1	在0时刻复位
186	
16.18.2	时钟沿之后复位
186	
16.18.3	创建仿真时钟
186	
16.18.4	在无效沿输入激励
187	
第17章	层次结构
188	
17.1	模块
188	
17.1.1	模块定义
188	
17.1.2	模块实例
188	
17.2	参数
188	
17.2.1	参数声明
189	
17.2.2	参数调整
189	
17.2.3	参数传递
190	
17.2.4	参数依赖
192	
17.2.5	内部参数
193	
17.2.6	clog2
193	
17.2.7	指数**
194	
17.3	端口
194	

17.3.1 端口声明	194
17.3.2 端口连接	195
17.3.3 实数传递	196
17.4 Generate语句	196
17.4.1 Loop generate construct	197
17.4.2 Conditional generate construct	200
17.5 实例数组	201
17.6 层次名字	203
第18章 系统任务和函数	205
18.1 显示任务	205
18.1.1 显示和写出任务	205
18.1.2 探测任务	208
18.1.3 监控任务	209
18.2 文件读写	209
18.2.1 打开和关闭文件	209
18.2.2 文件输出	211
18.2.3 字符串输出	212
18.2.4 文件输入	213
18.2.5 文件定位	216
18.2.6 刷新输出	216
18.2.7 错误状态	216
18.2.8 检查文件尾部	217
18.2.9 加载文件数据	217
18.3 时间比例	218
18.3.1 \$printtimescale	

218	
18.3.2	\$timeformat
218	
18.4	仿真控制
218	
18.4.1	\$finish
218	
18.4.2	\$stop
218	
18.5	仿真时间
218	
18.6	转换函数
219	
18.7	概率分布
220	
18.7.1	\$random
220	
18.7.2	\$dist_functions
220	
18.8	命令行输入
220	
18.8.1	\$test\$plusargs
221	
18.8.2	\$value\$plusargs
221	
18.9	数学运算
223	
18.9.1	整数函数
223	
18.9.2	实数函数
223	
18.10	波形记录
224	
第19章	编译指令
225	
19.1	`celldefine和`endcelldefine
225	
19.2	`default_nettype
225	
19.3	`define和`undef
226	
19.4	`ifdef、`else、`elsif、`endif、`ifndef
227	
19.5	`include
228	
19.6	`resetall
228	
19.7	`line
228	

19.8	`timescale	229
19.9	`unconnected_drive和`nounconnected_drive	230
19.10	`begin_keywords和`end_keywords	230
19.11	`pragma	230
第20章 Specify块		
20.1	specify块声明	231
20.2	speparam	231
20.3	模块路径声明	232
20.3.1	模块路径要求	232
20.3.2	简单路径	232
20.3.3	沿敏感路径	233
20.3.4	状态依赖路径	234
20.4	模块路径延迟	235
第21章 时序检查		
21.1	概览	237
21.2	使用稳定窗口的时序检查	237
21.2.1	\$setup、\$hold、\$setuphold	238
21.2.2	\$recovery、\$removal、\$recrem	238
21.3	时钟和控制信号的时序检查	240
21.3.1	\$skew、\$timeskew、\$fullskew	240
21.3.2	\$width	240
21.3.3	\$period	241
21.3.4	\$nochange	241
21.4	使用notifier响应时序违反	241
21.5	使用条件事件	

242	
21.6	时序检查中的Vector
243	
21.7	Negative timing check
243	
	第22章 反标SDF
246	
22.1	SDF标注器
246	
22.2	SDF construct到Verilog的映射
246	
22.2.1	SDF路径延迟到Verilog的映射
246	
22.2.2	SDF时序检查到Verilog的映射
247	
22.2.3	SDF互连延迟的标注
248	
22.3	\$sdf_annotate
249	
22.4	SDF文件例子
250	
	第23章 编程语言接口
252	
23.1	DirectC
252	
23.2	SystemVerilog
252	
	第24章 综合指令
253	
24.1	Synopsys综合指令
253	
24.2	使用综合指令
253	
24.3	使用translate_off/on
254	
24.4	误用translate_off/on
256	
24.5	使用attribute
256	
	第三部分 书写文档
	第25章 书写文档
260	
25.1	文档格式
260	
25.2	定义文档
261	
25.3	应用文档
262	
25.4	设计文档

262	
25.5	备份文档
263	
25.6	GPIO设计
263	
第26章	GPIO应用文档
264	
26.1	Overview
264	
26.2	Register Description
264	
26.2.1	PIN Level Register (PIN)
265	
26.2.2	Data Register (DAT)
265	
26.2.3	Data Set Register (DATS)
265	
26.2.4	Data Clear Register (DATC)
265	
26.2.5	Mask Register (IM)
266	
26.2.6	Mask Set Register (IMS)
266	
26.2.7	Mask Clear Register (IMC)
266	
26.2.8	PULL Enable Register (PEN)
266	
26.2.9	PEN Enable Set Register Register (PENS)
266	
26.2.10	PEN Enable Clear Register Register (PENC)
266	
26.2.11	PSEL Select Register (PSEL)
266	
26.2.12	PSEL Enable Set Register Register (PSELS)
266	
26.2.13	PSEL Enable Clear Register Register (PSELC)
267	
26.2.14	Function Register (FUN)
267	
26.2.15	Function Set Register (FUNS)
267	
26.2.16	Function Clear Register (FUNC)
267	
26.2.17	Select Register (SEL)
267	
26.2.18	Select Set Register (SELS)
267	
26.2.19	Select Clear Register (SELC)
267	

26.2.20	Direction Register (DIR)	267
26.2.21	Direction Set Register (DIRS)	268
26.2.22	Direction Clear Register (DIRC)	268
26.2.23	Trigger Register (TRG)	268
26.2.24	Trigger Set Register (TRGS)	268
26.2.25	Trigger Clear Register (TRGC)	268
26.2.26	FLAG Register (FLG)	268
26.2.27	FLAG Clear Register (FLGC)	269
26.3	Program Guide	269
26.3.1	GPIO Function Guide	269
26.3.2	Alternate Function Guide	269
26.3.3	Interrupt Function Guide	269
26.3.4	Disable Interrupt Function Guide	270
第27章	GPIO设计文档	271
27.1	文件列表 (见表27-1)	271
27.2	端口列表 (见表27-2)	271
27.3	配置参数 (见表27-3)	272
第四部分	高级设计	
第28章	使用IP	274
28.1	Cadence的IP	274
28.2	Cadence的VIP	275
28.3	Synopsys的IP	275
28.4	DesignWare Building Block	276
28.5	在FPGA上使用DesignWare	276
第29章	代码优化	278

29.1 代码可读	278
29.2 简洁编码	279
29.3 优化逻辑	281
29.4 优化迟到信号	281
29.5 括号控制结构	282
第30章 状态机设计	283
30.1 状态机类型	283
30.2 状态编码方式	283
30.3 二进制编码FSM	284
30.3.1 两个always块	284
30.3.2 重要的编码规则	285
30.3.3 错误状态的转换	285
30.3.4 next的默认值	285
30.4 独热码编码FSM	286
30.5 寄存器输出	287
第31章 可配置设计	289
31.1 格雷码转换	289
31.2 通用串行CRC	290
31.2.1 general_crc.v	290
31.2.2 testbench	292
31.3 FIFO控制器	293
31.4 RAM Wrapper 例子	296
31.4.1 常规方法	296
31.4.2 名字规范化	297
31.4.3 RF1_wrapper.v	

298
31.4.4 gen_wrapper.pl
302
31.4.5 ram_def.txt例子
306
31.4.6 生成wrapper
307
31.5 可配置的GPIO设计
308
31.5.1 gpio.v
308
31.5.2 gpio_params.v
317
31.5.3 gpio_check.v
317
31.5.4 gpio_reg.v
318
31.5.5 gpio_sync.v
319
31.6 可配置的BusMatrix
320
31.6.1 BusMatrix简介
320
31.6.2 设计ABM
321
31.6.3 mini_abm
322
31.6.4 large_abm
331
31.7 可配置的Andes Core N801
333
31.8 可配置的ARM926EJS
334
31.9 灵活的coreConsultant
336
第32章 可测性设计
337
32.1 内部扫描
337
32.2 内建自测
339
32.3 边界扫描
340
第五部分 时钟和复位
第33章 异步时序
342
33.1 亚稳态
342
33.2 MTBF

343	
33.3	同步器
344	
33.3.1	电平同步器
344	
33.3.2	边沿检测同步器
345	
33.3.3	脉冲检测同步器
345	
33.4	同步多位数据
347	
33.5	异步FIFO
348	
33.6	Design Ware
348	
33.7	DW_fifoclt_s2_sf
349	
33.8	门级仿真
351	
第34章	时钟生成
352	
34.1	同步电路
352	
34.2	设计原则
353	
34.3	分频器
353	
34.3.1	1/n分频器
353	
34.3.2	n/d分频器
355	
34.4	时钟切换
355	
34.5	时钟生成
358	
第35章	时钟例子
362	
35.1	Overview
362	
35.2	CGU Clock
362	
35.2.1	Clock List
362	
35.2.2	Clock Diagram (见图35-1)
363	
35.2.3	Clock Divider Rate (见表35-1)
364	
35.3	Register Description (见表35-2)
364	

35.3.1 CGU PLL Divider Register (CGU_PDR)	364
35.3.2 CGU Counter Register (CGU_CNT)	365
35.3.3 CGU PLL Control Register (CGU_PCR)	365
35.3.4 CGU Low Power Control Register (CGU_LPC)	365
35.3.5 CGU Status Register (CGU_CST)	365
35.3.6 CGU Divider 0 Register (CGU_DV0 1/s)	366
35.3.7 CGU Divider 1 Register (CGU_DV1 1/x)	366
35.3.8 CGU Divider 2 Register (CGU_DV2 1/n)	366
35.3.9 CGU Divider 3 Register (CGU_DV3 1/n)	367
35.3.10 CGU Divider 4/5/6/7 Register (CGU_DV4/5/6/7 n/d)	367
35.3.11 CGU Divider 8 Register (CGU_DV8 n/d)	367
35.3.12 CGU Divider 9 Register (CGU_DV9 n/d)	367
35.3.13 CGU Module Stop 0 Register (CGU_MS0)	367
35.3.14 CGU Module Stop 1 Register (CGU_MS1)	368
35.3.15 CGU Module Stop 2 Register (CGU_MS2)	368
35.3.16 CGU Reset Control Register (CGU_RCR)	369
35.3.17 CGU Reset Status Register (CGU_RST)	369
35.4 PLL Structure	369
35.4.1 Frequency Calculation	370
35.4.2 VCO Frequency Limitation	370
35.4.3 PFD Clock Frequency Limitation	370
35.5 PLL Control	371
35.6 Sleep and Wakeup	371
35.6.1 State switch	371
35.6.2 How to wakeup	

372	
35.7	Module Stop
372	
35.8	Application Notes
373	
第36章 复位设计	
374	
36.1	复位的用途
374	
36.2	寄存器编码风格
374	
36.2.1	有/无同步复位寄存器
374	
36.2.2	寄存器推导原则
376	
36.3	同步复位
376	
36.3.1	编码风格和电路
377	
36.3.2	同步复位的优点
378	
36.3.3	同步复位的缺点
379	
36.4	异步复位
379	
36.4.1	编码风格和电路
380	
36.4.2	既有异步复位又有异步置位的寄存器
380	
36.4.3	异步复位的优点
381	
36.4.4	异步复位的缺点
382	
36.5	异步复位的问题
382	
36.5.1	复位recovery时间
383	
36.5.2	复位撤销经历不同的时钟周期
383	
36.6	复位同步器
383	
36.6.1	复位同步器有亚稳态吗？
384	
36.6.2	错误的ASIC Vendor模型
385	
36.6.3	有缺点的复位同步器
385	
36.6.4	复位时的仿真验证
386	

36.7 复位分布树	387
36.7.1 同步复位分布技巧	389
36.7.2 异步复位分布技巧	389
36.7.3 复位分布树的时序分析	390
36.8 复位毛刺的过滤	391
36.9 异步复位的DFT	391
36.10 多时钟复位的问题	392
36.10.1 非协调的复位撤销	392
36.10.2 顺序协调的复位撤销	393
36.11 结论	394
第六部分 验证之路	
第37章 验证之路	396
37.1 整洁验证	397
37.2 验证目标	398
37.3 验证流程	398
37.4 验证计划	398
37.5 随机验证	399
37.6 直接验证	399
37.7 白盒验证	399
37.8 模块验证	400
37.9 系统验证	400
37.9.1 验证重点	400
37.9.2 验证环境	401
37.9.3 IP互连	401
37.9.4 性能验证	401

37.10 DFT验证	402
37.11 网表验证	402
37.12 高级抽象	403
37.13 灵活验证	405
37.14 ARM926EJS的Validation环境	406
37.14.1 Validation tools	407
37.14.2 Validation configuration files	407
37.14.3 Validation test suites	407
37.14.4 Validation flow	408
37.14.5 Building the model	408
37.14.6 Running Validation test suites	408
37.14.7 Debugging a single Validation test	410
37.15 AHB BusMatrix的验证	411
37.16 某芯片的SoC验证环境	411
第七部分 其他介绍	
第38章 SystemVerilog特性	414
38.1 SystemVerilog与Systemc比较	414
38.2 SystemVerilog的特点	414
38.3 新的数据类型	415
38.3.1 整型和实型	415
38.3.2 新的操作符	416
38.3.3 数组	416
38.3.4 队列	417
38.3.5 枚举类型	417
38.3.6 结构体和共同体	417

38.4	always_comb、always_latch和always_ff	417
38.5	unique和priority	418
38.6	loop、break和continue	419
38.7	task和function	419
38.7.1	静态和自动作用域	419
38.7.2	参数传递	420
38.7.3	参数中的默认值	420
38.8	Port connection	421
38.9	Tag	421
38.10	Interface	422
38.11	class和object	425
38.11.1	对象的概念	425
38.11.2	类的创建	426
38.11.3	类的继承	427
38.11.4	类的randomize	428
38.11.5	类的cover group	429
38.12	VMM、OVM和UVM	429
	参考文献	431
	关于版权	432

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com