

УДК 004.438
ББК 32.973.2
Т56

Дональд Томас

Т56 Логическое проектирование и верификация систем на SystemVerilog / пер. с англ. А. А. Слинкина, А. С. Камкина, М. М. Чупилко; науч. ред. А. С. Камкин, М. М. Чупилко. – М.: ДМК Пресс, 2019. – 384 с.: ил.

ISBN 978-5-97060-619-3

Книга посвящена SystemVerilog – языку описания аппаратуры, используемому для моделирования электронных систем. Разработчики SystemVerilog сделали его синтаксис похожим на синтаксис языка С, что упрощает освоение. Предполагается, что у читателя есть базовая подготовка в области схемотехники и программирования. Материал по языку дается вместе с материалом по логическому проектированию, так что книга может использоваться в качестве учебного пособия для курсов цифровой схемотехники и архитектуры компьютеров. В современных подходах к проектированию аппаратуры проверка модели (верификация) не менее важна, чем ее разработка. SystemVerilog предлагает конструкции, позволяющие лучше отразить инженерный замысел в моделях, программные абстракции, упрощающие разработку тестовых окружений, утверждения, обеспечивающие проверку поведения сложных систем, а также средства измерения функционального покрытия в процессе верификации.

Издание будет полезно студентам, проходящим вводный курс цифровой схемотехники, а также разработчикам, которые знакомы с Verilog или VHDL, но желают освежить свои навыки или нуждаются в кратком справочнике по SystemVerilog.

УДК 004.438
ББК 32.973.2

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the author, except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, or computer software is forbidden

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

ISBN 978-1-52336-402-2 (анг.)
ISBN 978-5-97060-619-3 (рус.)

© 2016 Donald Thomas
© Оформление, издание, перевод, ДМК Пресс, 2019

Оглавление

Предисловие: об этой книге	13
Предисловие от издательства	15
Контекст: проектирование на уровне регистровых передач	16
Благодарности	18
Глава 1. Введение	19
1.1. Приступая к работе	20
1.1.1. Структурное описание	20
1.1.2. Как интерпретируется описание модуля	22
1.2. Моделирование цифровых систем	24
1.2.1. Замечания по поводу симуляции	25
1.2.2. Что делает симулятор?	26
1.2.3. Более подробно о симуляции	28
1.2.4. Модели исполнения SystemVerilog	32
1.2.5. Зачем все это?	33
1.3. Иерархия модулей	33
1.4. Тестовое окружение для модуля <code>mux</code>	35
1.4.1. Простой пример	35
1.4.2. Более интеллектуальное тестовое окружение	38
1.5. Резюме	40
1.6. Задачи и упражнения	40
Часть I. МОДЕЛИ УРОВНЯ РЕГИСТРОВЫХ ПЕРЕДАЧ	43
Глава 2. Комбинационные схемы	45
2.1. Моделирование комбинационных схем	45
2.1.1. Операторы <code>assign</code> и <code>always_comb</code>	46
2.1.2. Вы уверены, что это комбинационные схемы?	49
2.2. Использование операторов <code>assign</code> и <code>always_comb</code>	50
2.2.1. Оператор <code>always_comb</code>	50
2.2.2. Оператор <code>assign</code>	51
2.2.3. Процедурное моделирование с помощью <code>if</code> и <code>case</code>	52
2.2.4. Задание несущественных комбинаций с помощью <code>unique case</code>	55
2.2.5. Упрощение спецификации с помощью <code>?</code> и <code>casez</code>	57
2.2.6. Моделирование с учетом уровней сигналов	58
2.2.7. Оператор <code>priority case</code>	59
2.3. Основы разработки тестового окружения	60
2.3.1. Отладочная печать	61
2.3.2. Основы тестирования комбинационных схем	62
2.3.3. Более сложная система	64
2.4. Параметризованные модули	66

2.5. Спецификация портов	68
2.6. Основные типы данных	70
2.6.1. Двух- и четырехзначные «биты»	70
2.6.2. Целочисленные типы данных	71
2.6.3. Перечисления и определения типов	72
2.6.4. Тестирование комбинационных схем с перечислениями	76
2.6.5. Структуры	77
2.7. Множественные драйверы	79
2.7.1. Цепи	80
2.7.2. Трестабильные приемопередатчики	81
2.8. Задачи и упражнения	82
Глава 3. Конечные автоматы	87
3.1. D-триггер	88
3.1.1. Смотри, куда ступаешь	89
3.1.2. Вариации на тему	90
3.1.3. Тестовое окружение для D-триггера	90
3.2. Основы проектирования конечных автоматов	92
3.2.1. Описание на SystemVerilog	93
3.2.2. Неблокирующие (параллельные) присваивания	94
3.2.3. Другой взгляд на = и <=	96
3.2.4. Наглядное изображение диаграмм состояний	99
3.2.5. Формальное определение	100
3.3. Явный стиль описания конечных автоматов	101
3.4. Логическая оптимизация	104
3.4.1. Кодирование состояний	104
3.4.2. Комбинационные схемы для функций перехода и выхода	105
3.4.3. Так ли вам нужны несущественные элементы?	106
3.5. Тестовые окружения для конечных автоматов	106
3.6. Задачи и упражнения	106
Глава 4. Предположение о синхронности	110
4.1. Основные предположения: доверяй, но проверяй	110
4.2. Предположения о временных характеристиках	111
4.2.1. Временные характеристики D-триггера	111
4.2.2. Расфазировка тактового сигнала	113
4.2.3. Нарушение ограничения на время удержания	114
4.3. Домены синхронизации	115
4.3.1. Что ограничивает размер домена синхронизации?	115
4.3.2. Междоменные сигналы	116
4.4. Логическая оптимизация: коррекция временных характеристик	118
4.5. Правила проектирования синхронных систем	119
Часть II. АППАРАТНЫЕ ПОТОКИ	123
Глава 5. Аппаратные потоки (конечные автоматы с трактом данных)	125

5.1. Аппаратные потоки	125
5.1.1. Иллюстративный пример	126
5.1.2. Временная диаграмма работы потока	127
5.1.3. Тракт данных потока	128
5.1.4. Диаграмма состояний	130
5.1.5. Совмещение конечного автомата и тракта данных	131
5.1.6. Описание на SystemVerilog	132
5.1.7. Формальное определение	133
5.2. Временные характеристики автоматов Мура и Мили	134
5.3. Компоненты тракта данных	135
5.3.1. Комбинационные элементы	136
5.3.2. Регистры	137
5.3.3. Дешифраторы	138
5.3.4. Шины	140
5.3.5. Модули памяти	141
5.4. Тестовые окружения для аппаратных потоков	143
5.5. Задачи и упражнения	143

Глава 6. Интерфейсы 153

6.1. Взаимодействующие аппаратные потоки	153
6.1.1. Организация потоков	153
6.1.2. Синхронность	155
6.2. Синхронные взаимодействия между потоками	156
6.2.1. Двустороннее ожидание	158
6.2.2. Пошаговая синхронизация	160
6.3. Пример шины SimpleBus	162
6.3.1. Определение протокола SimpleBus	162
6.3.2. Поток интерфейса процессора (ведущий компонент)	165
6.3.3. Поток интерфейса памяти (ведомый компонент)	167
6.3.4. Система в целом	169
6.3.5. Код примера SimpleBus	171
6.4. Асинхронные взаимодействия между потоками	176
6.4.1. Протокол квитирования с полной взаимной синхронизацией	177
6.4.2. Вариации на тему	180
6.4.3. Очереди как буферы	181
6.5. Интерфейсы в SystemVerilog	183
6.5.1. Пример простого интерфейса	184
6.5.2. Характеристики интерфейса	186
6.5.3. Пример более сложного интерфейса	187
6.6. Задачи и упражнения	192

Часть III. ТЕСТОВЫЕ ОКРУЖЕНИЯ 197

Глава 7. Введение в тестовые окружения 199

7.1. Организация тестового окружения	199
7.2. Программы тестового окружения	200
7.2.1. Конструкция program	201

7.2.2. Этапы работы симулятора.....	202
7.3. Тестовые окружения для конечных автоматов	204
7.3.1. Тактовый сигнал и сигнал сброса	204
7.3.2. Использование \$monitor для отладки конечных автоматов.....	206
7.3.3. Неявно заданные конечные автоматы	208
7.4. Тестовые окружения для аппаратных потоков	213
7.4.1. Запуск аппаратного потока	215
7.4.2. Системная инициализация	217
7.4.3. Простая программа тестового окружения	217
7.4.4. Использование процедур	219
7.4.5. Использование классов.....	221
7.4.6. Обратите внимание	224
7.5. Использование случайных значений.....	225
7.5.1. Определение случайной переменной	225
7.5.2. Ограничения на случайные значения	226
7.5.3. Случайный выбор	229
7.5.4. Случайные последовательности	230
7.6. Полезные конструкции	234
7.6.1. Иерархические имена.....	234
7.6.2. Операторы force/release и assign/deassign.....	236
7.6.3. Завершение симуляции.....	238
7.7. Отладка с использованием процедур ввода-вывода	239
7.7.1. Процедуры \$display, \$monitor и \$strobe.....	239
7.7.2. Контроль выводимых величин.....	240
7.7.3. Файловый ввод/вывод.....	241
7.7.4. Процедуры \$readmemb и \$readmemh.....	241
7.8. Задачи и упражнения	242

Глава 8. Параллельные тестовые окружения 244

8.1. Процессы	244
8.2. Пример и общая схема тестирования.....	246
8.3. Протоколы взаимодействия.....	248
8.4. Организация тестового окружения	249
8.4.1. Заголовки и создание экземпляров модулей	249
8.4.2. Тестовый передатчик	250
8.4.3. Тестовый приемник.....	253
8.4.4. Основная часть тестового окружения	255
8.5. Конструкции параллельного программирования	257
8.5.1. Оператор wait.....	257
8.5.2. Оператор disable	258
8.5.3. Почтовые ящики.....	259
8.5.4. Семафоры.....	261
8.5.5. Именованные события	263

Глава 9. Утверждения и последовательности 265

9.1. Предварительные сведения	266
9.1.1. Пример непосредственного утверждения.....	266
9.2. Введение в параллельные утверждения.....	268

9.2.1. Определение и проверка свойства	269
9.2.2. Свойства и последовательности	271
9.2.3. Как интерпретируются утверждения	272
9.2.4. Автоматный взгляд на последовательности	273
9.2.5. Еще о предыдущем примере	273
9.3. Последовательности с диапазонами и повторениями	276
9.3.1. Определение протокола SimpleBus	276
9.3.2. Спецификация свойств протокола	277
9.3.3. Операции над последовательностями	279
9.3.4. Повторение частей в последовательностях	281
9.4. Вычисления внутри последовательностей	284
9.5. Функции работы с сэмплированными значениями	287
9.5.1. Общая информация	287
9.5.2. Использование в последовательностях	289
9.6. Задачи и упражнения	292

Глава 10. Функциональное покрытие 293

10.1. План верификации	293
10.2. Группы покрытия и точки покрытия	294
10.2.1. Иллюстрация	295
10.2.2. Параметры накопителей	298
10.2.3. Активация групп покрытия в утверждениях	299
10.3. Покрытие переходов и перекрестное покрытие	301
10.3.1. Покрытие переходов	301
10.3.2. Накопители переходов	302
10.3.3. Накопители для перекрестного покрытия	304
10.4. Вычисление уровня покрытия	305
10.5. Задачи и упражнения	307

Часть IV. ДЕТАЛИ, ДЕТАЛИ, ДЕТАЛИ 309

Глава 11. Процедурные модели 311

11.1. Операторы процессов	311
11.1.1. Оператор initial	313
11.1.2. Операторы always_comb и always_latch	314
11.1.3. Оператор always_ff	316
11.1.4. Оператор непрерывного присваивания (assign)	316
11.1.5. Оператор final	317
11.2. Оператор if-else и условная операция	317
11.2.1. Логические выражения (условия)	318
11.2.2. Логические связки в выражениях	319
11.2.3. Многовариантное ветвление с помощью if-else-if	320
11.3. Оператор case и его разновидности	321
11.3.1. Операторы casex и casez	323
11.3.2. Ключевые слова unique, unique0 и priority	324
11.4. Циклы	324
11.4.1. Цикл forever	325

12 ❖ Оглавление

11.4.2. Цикл repeat	325
11.4.3. Цикл while	325
11.4.4. Цикл do-while	326
11.4.5. Цикл for.....	326
11.4.6. Операторы continue и break	327
11.4.7. Цикл foreach.....	327
11.5. Подпрограммы: функции и процедуры	328
11.5.1. Функции	329
11.5.2. Процедуры.....	330
11.5.3. Сходства и различия.....	332
11.5.4. Направление и тип аргументов	332
11.5.5. Возвращаемое значение	332
11.5.6. Автоматические и статические переменные	334
11.5.7. Значения аргументов по умолчанию	335
11.6. Таблица операций	336

Глава 12. Структурные модели..... 338

12.1. Вентильные примитивы	338
12.2. Цепи	341
12.2.1. Объявление цепей	343
12.2.2. Установка значений в цепях	344
12.3. Выбор части и конкатенация	346
12.3.1. Выбор бита и выбор части.....	346
12.3.2. Конкатенация и репликация.....	347
12.4. Модули, порты и экземпляры модулей	349
12.4.1. Модули и их экземпляры	349
12.4.2. Порты модулей.....	351
12.5. Генерация моделей	353

Глава 13. Массивы..... 356

13.1. Массивы	356
13.1.1. Многомерные массивы	357
13.1.2. Упакованные и неупакованные массивы.....	359
13.1.3. Операции над массивами	359
13.2. Динамические массивы.....	360
13.3. Строки	361
13.4. Очереди.....	362
13.5. Ассоциативные массивы.....	363

Глава 14. Работа симулятора..... 365

14.1. События, слоты времени и списки событий	365
14.2. Цикл симуляции.....	366
14.3. Основной и реагирующий этапы	369
14.4. Блок-схема работы симулятора	372

Предметный указатель..... 374