

Оглавление

Предисловие	3
Глава 1. Введение	8
1.1. История языка SystemVerilog	9
1.2. Первоисточники языка SystemVerilog	9
1.3. Стандарты языка SystemVerilog	10
1.4. Введение в верификацию цифровых систем	11
1.4.1. Разработчики аппаратуры и инженеры по верификации	12
1.4.2. Взаимодействие тестируемого проекта и тестового стенда	12
1.4.3. Проблемы тестирования на различных уровнях проектирования	13
1.5. Тестовый стенд	14
1.5.1. Плоский тестовый стенд	15
1.5.2. Многоуровневый тестовый стенд	17
1.5.3. Функциональные модели шин	19
1.6. Утверждения	19
1.7. Функциональное покрытие	20
1.8. Рандомизация	22
1.8.1. Направленное тестирование	22
1.8.2. Тестирование на основе рандомизации	22
1.8.3. Ограниченное случайное управляющее воздействие	23
1.8.4. Объекты рандомизации	23
1.8.5. Начальные значения в рандомизации	25
1.9. Выполнение моделирования	25
1.9.1. Введение в событийное моделирование	25
1.9.2. Время моделирования, временной интервал, гонки	25
1.9.3. Регионы событий	26
1.9.4. Алгоритм моделирования	29
1.9.5. Статическая формальная верификация и логические конусы	30
1.10. Системные задачи отображения информации	31
1.10.1. Задачи отображения и записи	31
1.10.2. Спецификации формата	32
1.10.3. Размер отображаемых данных	33
1.10.4. Неизвестные значения и значения высокого импеданса	34
1.10.5. Формат шаблона назначений	35
1.10.6. Задачи стробированного мониторинга	36

1.10.7. Задачи непрерывного мониторинга	36
1.10.8. Системная задача формата времени <code>\$timeformat</code>	37
Глава 2. Типы данных	40
2.1. Основные типы данных	40
2.1.1. Сети	41
2.1.2. Переменные	47
2.1.3. Целочисленные типы данных	50
2.1.4. Типы данных действительных чисел	51
2.1.5. Строковый тип данных	52
2.1.6. Тип данных событие	55
2.2. Массивы фиксированного размера	55
2.2.1. Объявление массивов	55
2.2.2. Многомерные массивы	58
2.2.3. Инициализация массивов при объявлении	60
2.2.4. Обращение к элементам, частям и фрагментам массивов	61
2.2.5. Присваивание значений массивам	62
2.2.6. Реализация циклов по элементам массивов	63
2.2.7. Операции над массивами	64
2.3. Другие виды массивов	65
2.3.1. Динамические массивы	65
2.3.2. Очереди	66
2.3.3. Ассоциативные массивы	68
2.4. Системные функции и методы для работы с массивами	70
2.4.1. Системные функции	71
2.4.2. Методы	73
2.4.3. Построение табло с помощью методов поиска	76
2.5. Выбор способа хранения данных	77
2.5.1. Гибкость	77
2.5.2. Использование памяти	77
2.5.3. Скорость	78
2.5.4. Доступ к данным	79
2.5.5. Выбор оптимальной структуры данных	79
2.6. Пользовательские типы	80
2.7. Структуры	81
2.7.1. Объявление структур	81
2.7.2. Присваивание значений структурам	82
2.7.3. Распакованные и упакованные структуры	84
2.8. Объединения	85
2.8.1. Объявление объединений	85
2.8.2. Распакованные и упакованные объединения	86
2.8.3. Обращение к элементам объединения	86
2.8.4. Теговые объединения	87
2.9. Перечисляемые типы	89
2.9.1. Объявление переменных перечисляемого типа	90

2.9.2. Автоматическая генерация меток перечисляемого типа	91
2.9.3. Значения перечисляемого типа	91
2.9.4. Выполнение операций на переменных перечисляемого типа ..	92
2.9.5. Системные методы для перечисляемых типов	93
2.9.6. Печать перечисляемых типов	94
2.9.7. Приведение выражений к перечисляемому типу	95
2.10. Константы	95
2.10.1. Константы <code>parameter</code> и <code>localparam</code>	96
2.10.2. Константы <code>specparam</code>	97
2.10.3. Константы <code>const</code>	97
2.10.4. Знак \$ как значение параметра	98
2.11. Приведение типов	98
2.11.1. Статическая операция приведения типов	98
2.11.2. Динамическая системная функция приведения типов <code>\$cast</code>	99
2.11.3. Приведение битовых потоков	101
2.11.4. Операция <code>type</code>	102
2.11.5. Системные функции преобразования типов	103
2.12. Пакеты	103
2.12.1. Определение пакетов	104
2.12.2. Ссылки на содержимое пакетов	105
2.12.3. Условная компиляция пакетов	107
2.13. Передача данных через порты и в качестве аргументов под- программ	108
2.13.1. Передача значений через порты модулей и интерфейсов	108
2.13.2. Передача в качестве аргументов подпрограмм массивов, структур и объединений	109
Глава 3. Выражения, операции и операторы назначения	111
3.1. Выражения	111
3.1.1. Константные выражения	111
3.1.2. Агрегатные выражения	112
3.1.3. Знаковые и беззнаковые арифметические выражения	112
3.1.4. Использование в выражениях операции назначения	113
3.1.5. Выражения строковых литералов	114
3.1.6. Размер битовых выражений	115
3.2. Операции	116
3.2.1. Приоритет операций	117
3.2.2. Операции с 2-мя и с 4-мя состояниями	117
3.2.3. Операции назначения	118
3.2.4. Операции инкремента и декремента	118
3.2.5. Операции равенства с подстановочными знаками	119
3.2.6. Логические операции	119
3.2.7. Операции сдвига	120
3.2.8. Конкатенация строк	120
3.2.9. Операция членства в множестве	121

3.2.10. Операции с битовыми потоками	122
3.3. Операторы назначения	123
3.4. Непрерывные назначения	124
3.5. Процедурные назначения	126
3.5.1. Оператор блокирующего назначения (=)	126
3.5.2. Оператор неблокирующего назначения (<=)	128
3.6. Управление временем в процедурных операторах назначения	129
3.6.1. Управление временем в операторах блокирующего назначения	129
3.6.2. Внутренние задержки в операторах блокирующего назначения	129
3.6.3. Управление временем в операторах неблокирующего назначения	130
3.6.4. Внутренние задержки в операторах неблокирующего назначения	131
3.7. Процедурные непрерывные назначения	131
Глава 4. Процессы и потоки	133
4.1. Процессы	133
4.2. Управление процедурным временем	134
4.2.1. Оператор управления задержками #	135
4.2.2. Оператор управления событиями @	135
4.2.3. Оператор ожидания wait	138
4.2.4. Управление процедурным временем внутри назначения	138
4.3. Процедурные блоки	140
4.3.1. Последовательный блок begin-end	141
4.3.2. Параллельные блоки (fork-join , join_any и join_none)	142
4.3.3. Время начала и время окончания процедурных блоков	144
4.3.4. Имена процедурных блоков	145
4.3.5. Метки процедурных операторов	146
4.4. Структурные процедуры	146
4.4.1. Процедура initial	147
4.4.2. Процедуры always	148
4.4.3. Процедура final	154
4.5. Управление процессами	154
4.5.1. Оператор wait fork	155
4.5.2. Оператор wait_order	156
4.5.3. Оператор disable	156
4.5.4. Оператор disable fork	158
4.5.5. Тонкое управление процессами	159
4.6. Потоки	160
4.7. Управление потоками	161
4.7.1. Блоки fork-join	161
4.7.2. Блоки fork-join и begin-end	162
4.7.3. Использование блока fork-join_none	163

4.7.4. Использование блока <code>fork-join_any</code>	163
4.7.5. Создание потоков в классе	164
4.7.6. Динамические потоки	165
4.7.7. Автоматические переменные в потоках	166
4.7.8. Ожидание всех порожденных потоков	168
4.7.9. Совместное использование переменных в потоках	168
4.8. Отключение потоков	169
4.8.1. Отключение одного потока	170
4.8.2. Отключение нескольких потоков	170
4.8.3. Отключение задачи, которая вызывается несколько раз	171
Глава 5. Процедурные операторы	173
5.1. Условный оператор <code>if-else</code>	174
5.1.1. Конструкция <code>if-else-if</code>	176
5.1.2. Квалификаторы <code>unique</code> , <code>unique0</code> и <code>priority</code>	177
5.2. Оператор <code>case</code>	179
5.2.1. Операторы <code>casez</code> и <code>casex</code>	181
5.2.2. Константные выражения в операторах <code>case</code>	182
5.2.3. Квалификаторы <code>unique</code> , <code>unique0</code> и <code>priority</code> в операторах <code>case</code>	183
5.2.4. Атрибуты <code>full_case</code> и <code>parallel_case</code>	184
5.2.5. Атрибут <code>full_case</code>	185
5.2.6. Атрибут <code>parallel_case</code>	186
5.2.7. Сравнение квалификаторов <code>unique</code> и <code>priority</code> с атрибутами <code>full_case</code> и <code>parallel_case</code> и конструкциями <code>else</code> и <code>default</code>	188
5.2.8. Квалификатор <code>inside</code> оператора <code>case</code>	189
5.3. Операторы циклов	189
5.3.1. Цикл <code>for</code>	189
5.3.2. Цикл <code>repeat</code>	191
5.3.3. Цикл <code>foreach</code>	192
5.3.4. Цикл <code>while</code>	193
5.3.5. Цикл <code>do-while</code>	193
5.3.6. Цикл <code>forever</code>	193
5.4. Операторы переходов	194
Глава 6. Модули	196
6.1. Определение модулей	196
6.2. Стили и уровни описания проектов	198
6.2.1. Структурный стиль описания проектов	198
6.2.2. Поведенческий стиль описания проектов	199
6.2.3. Уровни описания проекта	200
6.3. Элементы модулей	200
6.4. Объявления портов	201
6.4.1. Форматы объявления портов ANSI и non-ANSI	201
6.5. Экземпляры модулей	204
6.6. Неявные соединения портов	206

6.6.1. Нотация <code>.name</code> неявного соединения портов	206
6.6.2. Нотация <code>.*</code> неявного соединения портов	208
6.7. Псевдонимы сетей	209
6.7.1. Оператор <code>alias</code>	209
6.7.2. Неявные объявления сетей в операторе <code>alias</code>	210
6.8. Использование псевдонимов сетей с нотациями неявных соединений <code>.name</code> и <code>.*</code>	211
6.9. Передача значений через порты модулей	212
6.10. Параметризованные модули	213
6.10.1. Объявление параметризованных модулей	213
6.10.2. Создание экземпляров параметризованных модулей	214
6.10.3. Параметризованные типы	214
6.11. Внешние модули (прототипы модулей)	215
Глава 7. Задачи и функции	218
7.1. Общая информация	218
7.2. Автоматические и статические задачи и функции	220
7.3. Задачи	221
7.3.1. Объявление задач	221
7.3.2. Вызов задач	222
7.4. Функции	223
7.4.1. Объявление функций	223
7.4.2. Возвращаемые значения	224
7.4.3. Аргументы функций типа <code>output</code> и <code>inout</code>	225
7.4.4. Функции типа <code>void</code>	226
7.4.5. Рекурсивные функции	226
7.5. Константные функции	227
7.6. Вызовы подпрограмм и передача аргументов	228
7.6.1. Передача аргументов по значению	229
7.6.2. Передача аргументов по ссылке	229
7.6.3. Значения аргументов по умолчанию	230
7.6.4. Передача аргументов по имени	231
7.6.5. Передача в качестве аргументов массивов, структур и объединений	232
7.7. Пустые задачи и функции	232
7.8. Параметризованные задачи и функции	232
7.9. Сравнение задач и функций	233
7.10. Системные задачи и функции	234
Глава 8. Интерфейсы	237
8.1. Общая информация	237
8.2. Объявление интерфейсов	238
8.2.1. Группирование сигналов с помощью интерфейсов	238
8.2.2. Порты интерфейсов	239
8.2.3. Использование нотаций <code>.name</code> и <code>.*</code>	241
8.2.4. Глобальные и локальные интерфейсы	241

8.3. Использование интерфейсов в качестве портов модулей	242
8.4. Создание экземпляров интерфейсов и подключение интерфейсов к портам модуля	243
8.4.1. Создание экземпляров интерфейсов	243
8.4.2. Подключение интерфейсов к портам модулей	243
8.4.3. Подключение интерфейсов к другим интерфейсам	243
8.5. Ссылка на сигналы в пределах интерфейса	244
8.6. Модпорты интерфейсов	245
8.6.1. Определения модпортов	245
8.6.2. Два способа подключения интерфейсов с модпортами к экземплярам модуле	246
8.6.3. Ограничение доступа к сигналам интерфейса с помощью модпортов	247
8.7. Использование в интерфейсах задач и функций	248
8.7.1. Интерфейсные методы	248
8.7.2. Импорт интерфейсных методов в модпорты	249
8.7.3. Экспорт интерфейсных методов	251
8.8. Использование в интерфейсах процедурных блоков	253
8.9. Параметризованные интерфейсы	253
8.10. Выражения модпортов	254
Глава 9. Классы	256
9.1. Объявление класса	258
9.1.1. Синтаксис объявления класса	258
9.1.2. Объявление классов в пакете	260
9.1.3. Объявление класса с <code>typedef</code>	260
9.2. Создание объектов класса	261
9.2.1. Дескриптор класса и дескриптор объекта	261
9.2.2. Конструкторы классов	262
9.2.3. Дескрипторы и объекты	263
9.2.4. Освобождение памяти для объектов	264
9.2.5. Вызов типизированного конструктора	264
9.3. Обращение к свойствам и методам объекта	265
9.3.1. Обращение к свойствам объекта	265
9.3.2. Обращение к методам объекта	266
9.3.3. Дескриптор <code>this</code>	266
9.3.4. Константные свойства класса	267
9.4. Работа с динамическими объектами	268
9.4.1. Передача объектов и дескрипторов методам	268
9.4.2. Изменение дескриптора в методе	269
9.4.3. Изменение объектов «в полете»	269
9.4.4. Массивы дескрипторов	270
9.5. Статические свойства и статические методы класса	271
9.5.1. Статические свойства класса	271
9.5.2. Статические методы класса	271

9.6. Вложенные классы	271
9.7. Копирование объектов	272
9.7.1. Копирование объекта с помощью оператора new (неглубокое копирование)	273
9.7.2. Написание функции глубокого копирования	273
9.8. Наследование классов	275
9.8.1. Базовые классы и подклассы	275
9.8.2. Обращение к элементам базового класса	276
9.8.3. Скрытие и инкапсуляция данных	276
9.8.4. Понижение — приведение дескриптора базового класса к указателю на объект производного класса	277
9.8.5. Виртуальные методы	279
9.8.6. Абстрактные (виртуальные) классы и чистые виртуальные методы	281
9.8.7. Полиморфизм	282
9.9. Объявления методов вне класса	283
9.10. Параметризованные классы и специализация	284
9.10.1. Параметризованные классы	284
9.10.2. Специализация класса	284
9.11. Оператор разрешения области действия класса (::)	286
9.11.1. Общая информация	286
9.11.2. Оператор :: для параметризованных классов	288
9.12. Интерфейсные классы	289
9.12.1. Синтаксис интерфейсного класса	289
9.12.2. Реализация интерфейсного класса	290
9.12.3. Расширение по сравнению с реализацией	291
9.12.4. Параметры и определение типов в интерфейсном классе ...	292
9.12.5. Приведение и назначение ссылки объекта	292
9.12.6. Ромбовидные отношения	293
9.12.7. Специализация параметризованного интерфейсного класса .	294
9.12.8. Частичная реализация	294
9.13. Примеры использования классов	295
9.13.1. Структура упрощенного тестового стенда	295
9.13.2. Базовая транзакция	295
9.13.3. Расширение класса транзакций для ввода ошибок	296
9.13.4. Класс драйвера	297
9.13.5. Класс простого генератора	298
9.13.6. Шаблон концептуального проекта при создании генератора	298
9.13.7. Класс окружающей среды	300
9.13.8. Простой тестовый стенд	301
Глава 10. Синхронизация моделирования и блоки синхронизации	302
10.1. Введение в синхронизацию процесса моделирования	302
10.1.1. Взаимодействие проекта и тестового стенда	302

10.1.2. Регионы событий	303
10.1.3. Определение задержек между проектом и тестовым стендом	304
10.2. Блоки синхронизации	305
10.2.1. Объявление блока синхронизации	306
10.2.2. Входные и выходные перекося	308
10.2.3. Иерархические выражения, сигналы и контекст блока синхронизации	309
10.2.4. Программы и блоки синхронизации	310
10.2.5. Интерфейсы и блоки синхронизации	311
10.3. События блока синхронизации	312
10.4. Синхронизация по умолчанию	313
10.5. Оператор задержки цикла <code>##</code>	313
10.6. Глобальная синхронизация	314
10.7. Выборка входов	315
10.8. Синхронные приводы (синхронная установка выходов)	316
10.8.1. Приводы и неблокирующие назначения	317
10.8.2. Управление выходными сигналами синхронизации	318
10.9. Генератор синхросигналов	319
10.10. Примеры использования синхронизации	320
10.10.1. Простой арбитр с тестовым стендом	320
10.10.2. Работа с событиями синхронизации	322
10.10.3. Управление сигналами интерфейса через блок синхронизации	323
10.10.4. Использование в интерфейсе двунаправленных сигналов ..	325
Глава 11. Программы	326
11.1. Синтаксис программ	326
11.2. Функционирование соединений портов программы при отсутствии блока синхронизации	328
11.3. Устранение гонок тестового стенда	329
11.4. Вызов программ из модулей и вызов задач и функций из программ	329
11.5. Анонимные программы	330
11.6. Завершение программ	330
11.7. Пример использования программы	331
Глава 12. Утверждения	333
12.1. Введение	333
12.1.1. Происхождение утверждений языка SystemVerilog	333
12.1.2. Что такое утверждение?	334
12.1.3. Преимущества использования утверждений	335
12.2. Немедленные утверждения	336
12.3. Отложенные утверждения	338
12.3.1. Действия отложенных утверждений	340
12.3.2. Точки очистки отложенных утверждений	341
12.3.3. Отключения отложенных утверждений	342

12.4. Параллельные утверждения	342
12.4.1. Параллельные утверждения с булевыми выражениями	344
12.4.2. Операция импликации	345
12.4.3. Основы синхронизации	346
12.4.4. Многопоточность параллельных утверждений	351
12.4.5. Функции выбора значений	352
12.4.6. Формальные аргументы	358
12.4.7. Оператор отключения свойств <code>disable iff</code>	360
12.4.8. Уровни серьезности	361
12.4.9. Оператор <code>bind</code> связывания свойств	361
12.4.10. Различия между последовательностями и свойствами	363
12.4.11. Операторы параллельных утверждений	364
12.5. Последовательности	369
12.5.1. Объявление последовательностей	372
12.5.2. Операции над последовательностями	374
12.5.3. Вызов подпрограмм из последовательностей	396
12.5.4. Последовательности как формальный аргумент	397
12.5.5. Последовательность как предшественник	398
12.5.6. Последовательности в списке чувствительности	399
12.5.7. Подпоследовательности (последовательность в последова- тельности)	399
12.5.8. Пустая последовательность <code>[*0]</code>	400
12.5.9. Сильные и слабые последовательности	401
12.5.10. Функции выбора прошлых и будущих значений	402
12.5.11. Конечная точка последовательности	404
12.5.12. Составление последовательностей из более простых после- довательностей	406
12.5.13. Вырожденные последовательности	407
12.6. Свойства	408
12.6.1. Объявление свойств	408
12.6.2. Свойство последовательности	410
12.6.3. Операции над свойствами	411
12.6.4. Рекурсивные свойства	422
12.7. Системные задачи и функции, используемые в утверждениях	424
12.7.1. Задачи управления утверждениями <code>\$assertoff</code> , <code>\$asserton</code> , <code>\$assertkill</code>	425
12.7.2. Задачи управления блоками активности утверждений <code>\$as- sertpasson</code> , <code>\$assertpasson</code> , <code>\$assertfailoff</code> , <code>\$assertfailon</code> , <code>\$assertnonvacu- ous</code> , <code>\$assertvacuousoff</code>	426
12.7.3. Универсальная задача управления утверждением <code>\$assert- control</code>	427
12.7.4. Системные функции битовых векторов	428
12.8. Утверждения с множественными синхросигналами	430
12.8.1. Последовательности с множественными синхросигналами ..	430
12.8.2. Свойства с множественными синхросигналами	431

12.8.3. Потоки синхронизации	432
12.8.4. Конечная точка последовательности с множественными синхросигналами	434
12.8.5. Инициализация локальных переменных в случае множественных синхросигналов	436
12.9. Локальные переменные в последовательностях и свойствах .	437
12.9.1. Объявление локальных переменных и основы их использования	437
12.9.2. Локальная переменная как формальный аргумент	439
12.9.3. Локальные переменные и метод <code>triggered</code>	440
12.9.4. Локальные переменные с участием операций <code>or</code> , <code>and</code> и <code>intersect</code>	440
12.9.5. Примеры использования локальных переменных	442
12.10. Асинхронные утверждения	447
Глава 13. Функциональное покрытие	450
13.1. Введение	450
13.1.1. Методика верификации и функциональное покрытие	450
13.1.2. Типы покрытия	453
13.1.3. Стратегии функционального покрытия	455
13.1.4. Пример простого функционального покрытия	458
13.2. Группы покрытия	459
13.2.1. Синтаксис группы покрытия	459
13.2.2. Формальные и актуальные аргументы группы покрытия ...	460
13.2.3. Запуск групп покрытия	461
13.2.4. Группы покрытия в классе	462
13.3. Точки покрытия	463
13.4. Бины	465
13.4.1. Синтаксис объявления бинов	466
13.4.2. Определение бинов для значений	467
13.4.3. Бины с конструкцией <code>with</code>	469
13.4.4. Бины для переходов	469
13.4.5. Автоматическое создание бинов	472
13.4.6. Ограничение количества автоматически создаваемых бинов	472
13.4.7. Бины с подстановочными знаками	473
13.4.8. Игнорирование значений или переходов точки покрытия ...	474
13.4.9. Недопустимые значения или переходы точки покрытия	474
13.4.10. Покрытие конечного автомата	475
13.5. Перекрестное покрытие (покрытие пересечения)	475
13.5.1. Синтаксис перекрестного покрытия	476
13.5.2. Определение бинов перекрестного покрытия	477
13.6. Опции покрытия	483
13.6.1. Опции экземпляров покрытия	483
13.6.2. Опции типа группы покрытия	487
13.7. Методы покрытия	489

13.7.1. Использование методов покрытия	490
13.7.2. Переопределение пользователем метода <code>sample</code>	490
13.8. Системные задачи и функции покрытия	491
Глава 14. Рандомизация	492
14.1. Введение	492
14.1.1. Основные концепции рандомизации	492
14.1.2. Что необходимо рандомизировать	493
14.1.3. Использование классов при рандомизации	495
14.1.4. Проверка результата рандомизации	497
14.2. Случайные переменные	497
14.2.1. Модификатор <code>rand</code>	498
14.2.2. Модификатор <code>randc</code>	498
14.2.3. Массивы	498
14.2.4. Структуры	499
14.2.5. Объединения	500
14.2.6. Перечисления	500
14.3. Методы рандомизации	500
14.3.1. Метод <code>randomize()</code>	500
14.3.2. Методы <code>pre_randomize()</code> и <code>post_randomize()</code>	501
14.3.3. Пример использования метода <code>pre_randomize()</code>	502
14.4. Ограничения	503
14.4.1. Блоки ограничений	503
14.4.2. Наследование ограничений, абстрактный (виртуальный) класс и чистые ограничения	504
14.4.3. Оператор членства в наборе <code>inside</code>	505
14.4.4. Взвешенные распределения (оператор <code>dist</code>)	507
14.4.5. Ограничение уникальности (оператор <code>unique</code>)	508
14.4.6. Двухнаправленные ограничения	508
14.4.7. Оператор импликации (<code>-></code>)	509
14.4.8. Условный оператор <code>if-else</code>	510
14.4.9. Оператор эквивалентности (<code><-></code>)	511
14.4.10. Оператор цикла по элементам массива <code>foreach</code>	511
14.4.11. Оператор сокращения массива <code>with</code>	512
14.4.12. Глобальные ограничения	513
14.4.13. Упорядочивание случайных переменных (конструкция <code>solve-before</code>)	513
14.4.14. Статические ограничения	514
14.4.15. Функции в ограничениях	515
14.4.16. Защита от ограничений	516
14.4.17. Мягкие ограничения	518
14.4.18. Встроенные ограничения	519
14.5. Вероятности решений	520
14.5.1. Вероятности решений без ограничений	520
14.5.2. Использование импликации	520

14.5.3. Использование импликации и двунаправленных ограничений	521
14.5.4. Управление распределением с помощью конструкции solve-before	522
14.6. Управление рандомизацией	522
14.6.1. Управление случайными переменными (метод <code>rand_mode()</code>)	522
14.6.2. Управление ограничениями (метод <code>constraint_mode()</code>)	524
14.6.3. Управление встроенными случайными переменными (метод <code>randomize()</code>)	526
14.6.4. Проверка встроенных ограничений	526
14.6.5. Динамическая модификация ограничений	527
14.6.6. Допустимые ограничения	527
14.7. Рандомизация массивов	528
14.7.1. Размер массива	528
14.7.2. Сумма элементов массива	528
14.7.3. Ограничение отдельных элементов массива	530
14.7.4. Создание массива уникальных значений	530
14.7.5. Рандомизация массива дескрипторов	531
14.7.6. Пример использования рандомизации массива	532
14.8. Системные функции и методы рандомизации	533
14.8.1. Системная функция <code>\$urandom</code>	533
14.8.2. Системная функция <code>\$urandom_range()</code>	534
14.8.3. Метод <code>srandom()</code>	534
14.8.4. Метод <code>get_randstate()</code>	534
14.8.5. Метод <code>set_randstate()</code>	535
14.9. Системные функции генерации случайных чисел и вероятностного распределения	535
14.9.1. Функция <code>\$random</code>	535
14.9.2. Функции вероятностного распределения	536
14.10. Генераторы случайных чисел и стабильность случайности	536
14.10.1. Генераторы случайных чисел	537
14.10.2. Стабильность случайности	537
14.10.3. Свойства стабильности случайности	537
14.10.4. Стабильность потока	539
14.10.5. Стабильность объекта	540
14.10.6. Ручная загрузка рандомизации	541
14.11. Оператор случайного выбора <code>randcase</code>	542
14.12. Генерация случайных последовательностей	543
14.12.1. Веса случайности продукции	545
14.12.2. Операторы продукции <code>if-else</code>	545
14.12.3. Оператор продукции <code>case</code>	545
14.12.4. Оператор продукции <code>repeat</code>	546
14.12.5. Чередование продукции (оператор <code>randjoin</code>)	546
14.12.6. Прерывание продукции (операторы <code>break</code> и <code>return</code>)	547

14.12.7. Передача значений между продуктами	548
14.13. Методы рандомизации для построения тестовых стендов	550
14.13.1. Использование переменных в ограничениях	550
14.13.2. Использование неслучайных значений	551
14.13.3. Проверка значений с помощью ограничений	551
14.13.4. Рандомизация отдельных переменных	551
14.13.5. Выключение и включение ограничений	552
14.13.6. Использование встроенных ограничений	552
14.13.7. Использование внешних ограничений	552
14.13.8. Расширение класса	553
14.14. Пример использования рандомизации при построении тестового стенда	553
Глава 15. Контролеры	557
15.1. Объявление контролеров	558
15.2. Создание экземпляра контролера	560
15.3. Вывод сигналов синхронизации и сброса из контекста	561
15.4. Процедуры контролера	562
15.5. Группы покрытия в контролерах	563
15.6. Переменные контролера	564
15.6.1. Назначения переменных контролера	566
15.6.2. Рандомизация переменных контролера с предположениями	567
15.6.3. Планирование семантик	570
15.7. Функции в контролерах	571
15.8. Пример сложного контролера	571
Глава 16. Синхронизация процессов и связь между процессами	573
16.1. События	573
16.1.1. Инициирование события	574
16.1.2. Ожидание события	574
16.1.3. Устойчивый запуск события с помощью метода triggered ...	575
16.1.4. Указание последовательности событий с помощью оператора wait_order	576
16.1.5. Операции над событиями	577
16.2. Примеры использования событий	579
16.2.1. Блокировка процессов на фронте событий	579
16.2.2. Ожидание запуска события с помощью метода triggered	579
16.2.3. Использование события в цикле	580
16.2.4. Передача события подпрограмме в качестве аргумента	581
16.2.5. Ожидание нескольких событий	581
16.3. Семафоры	583
16.3.1. Методы работы с семафорами	584
16.3.2. Пример использования семафоров	584
16.4. Почтовые ящики	585
16.4.1. Методы работы с почтовыми ящиками	586
16.4.2. Параметризованные почтовые ящики	587

16.5. Примеры использования почтовых ящиков	588
16.5.1. Простое использования почтового ящика для передачи дан- ных между двумя объектами	588
16.5.2. Почтовый ящик в тестовом стенде	590
16.5.3. Ограниченные почтовые ящики	591
16.5.4. Несинхронизированные потоки, взаимодействующие с поч- товым ящиком	592
16.5.5. Синхронизация потоков с помощью ограниченного почтово- го ящика и метки <code>peek()</code>	594
16.5.6. Синхронизация потоков с помощью почтового ящика и со- бытия	595
16.5.7. Синхронизация потоков с помощью двух почтовых ящиков	596
Глава 17. Временные характеристики	599
17.1. Маршрут проектирования с временными характеристиками .	600
17.2. Модели задержек	601
17.2.1. Единицы измерения времени	601
17.2.2. Нотации задержек	602
17.2.3. Классификация задержек по соединению	603
17.3. Блок спецификации	604
17.3.1. Параметры спецификации	605
17.3.2. Объявление путей модуля	605
17.3.3. Простые пути модуля	605
17.3.4. Параллельное соединение	606
17.3.5. Полное соединение	606
17.3.6. Пути, чувствительные к фронту	607
17.3.7. Пути, зависящие от состояния	608
17.3.8. Объявление нескольких путей модулей в одном операторе ..	609
17.3.9. Полярность пути модуля	610
17.3.10. Назначение задержек путям модуля	611
17.3.11. Определение задержек переходов на путях модуля	611
17.4. Проверки синхронизации	613
17.4.1. Проверки синхронизации с использованием окна стабильно- сти	613
17.4.2. Проверки синхронизации для тактовых и управляющих сиг- налов	616
17.4.3. Спецификаторы управления фронтом	620
17.4.4. Уведомители	621
17.4.5. Включение проверок синхронизации	622
17.5. Стандартный формат задержек SDF	622
17.5.1. SDF в процессе проектирования	623
17.5.2. Раздел заголовка	624
17.5.3. Раздел ячеек	626
17.5.4. Задержки	626
17.5.5. Определение задержек	627
17.5.6. Проверки синхронизации	628

17.6. Обратное аннотирование	631
17.6.1. Отображение конструкций файла SDF в язык SystemVerilog	632
17.6.2. Множественные аннотации	635
17.6.3. Загрузка данных синхронизации из файла SDF	636
17.7. Пример использования временных характеристик при описа- нии пользовательского D-триггера	637
Глава 18. Защита исходного кода проекта от несанкционирован- ного доступа	640
18.1. Общая информация	640
18.2. Обработка защитных конвертов	641
18.2.1. Шифрование	641
18.2.2. Дешифрование	643
18.3. Директивы прагмы protect	643
18.4. Описание директив прагмы protect	643
Глава 19. Структура многоуровневого тестового стенда с пото- ками и межпроцессными связями	653
19.1. Транзактор (Agent)	653
19.2. Класс конфигурации	654
19.3. Класс среды	654
19.4. Программа тестового стенда	655
Заключение	657
Приложение А. Ключевые слова языка SystemVerilog	658
Литература	660
Список сокращений	661
Предметный указатель	662