

Оглавление

Предисловие	3
Глава 1. Встраиваемые системы и программируемые логические интегральные схемы	7
1.1. Встраиваемые системы	7
1.1.1. Понятие встраиваемой системы. Примеры. История. Область использования	7
1.1.2. Функции встраиваемых систем	8
1.1.3. Элементная база встраиваемых систем	8
1.1.4. Периферийные устройства	13
1.1.5. Надежность функционирования встраиваемых систем	13
1.1.6. Методология проектирования встраиваемых систем на ПЛИС	15
1.2. Программируемые логические интегральные схемы	18
1.2.1. Классификация ПЛИС	18
1.2.2. Программируемые логические устройства (PLD)	18
1.2.3. Сложные программируемые логические устройства (CPLD)	20
1.2.4. Программируемые пользователем вентильные матрицы (FPGA)	21
1.2.5. Системы на кристалле (SoC)	23
1.2.6. Использование PLD, CPLD, FPGA и SoC для построения встраиваемых систем	23
Выводы	24
Глава 2. Элементы языка описания аппаратуры Verilog	28
2.1. Модули	28
2.1.1. Объявление модулей	28
2.1.2. Элементы модулей	29
2.1.3. Поведенческий и структурный стили описания проектов	30
2.1.4. Пример одноразрядного сумматора	30
2.1.5. Объявления портов	33
2.1.6. Экземпляры модулей	35
2.1.7. Передача сигналов экземплярам модулей	35
2.1.8. Параметры модулей	37
2.1.9. Неявная передача значений параметров, оператор defparam	39
2.1.10. Массивы экземпляров модулей	39
2.1.11. Параллелизм языка Verilog	40
2.2. Базовые элементы языка Verilog	41
2.2.1. Ключевые слова	41
2.2.2. Идентификаторы	42
2.2.3. Белые знаки	42

2.2.4. Комментарии	42
2.2.5. Сигналы, сети, драйверы	43
2.2.6. Логические значения	43
2.2.7. Целые числа	43
2.2.8. Действительные числа	45
2.2.9. Примитивы языка Verilog	45
2.3. Типы данных	46
2.3.1. Два класса типов данных	46
2.3.2. Сети	47
2.3.3. Значение сигнала сети	48
2.3.4. Переменные	49
2.3.5. Параметры	51
2.3.6. Локальные параметры	51
2.3.7. Переменные генерации	52
2.3.8. Строки	52
2.3.9. Скаляры и векторы, векторы и массивы	52
2.3.10. Выбор битов и битовых полей	53
2.3.11. Выбор элементов массива и битовых полей элементов массива	53
2.3.12. Объявление памяти	54
2.4. Выражения	55
2.5. Операции языка Verilog	55
2.5.1. Побитовые операции	55
2.5.2. Операции редукции	57
2.5.3. Логические операции	58
2.5.4. Операции отношения	59
2.5.5. Операции идентичности	60
2.5.6. Арифметические операции	60
2.5.7. Разносторонние операции	60
2.5.8. Выполнение операций	62
2.5.9. Размеры битовых выражений	62
2.5.10. Приоритет операций	62
2.6. Оператор непрерывного назначения assign	63
2.6.1. Присваивание значений в языке Verilog	63
2.6.2. Форматы оператора непрерывного назначения	64
2.6.3. Использование оператора непрерывного назначения	65
2.7. Процедурные операторы и блоки	67
2.7.1. Процедурные операторы initial и always, процедурные блоки	67
2.7.2. Операторные скобки begin-and и fork-join	67
2.7.3. Именованные процедурные блоки	68
2.7.4. Формат процедурных блоков	68
2.8. Управление процедурным временем	70
2.8.1. Оператор чувствительности @	70
2.8.2. Список чувствительности	71
2.8.3. Список чувствительности в комбинационных схемах	72

2.8.4. Список чувствительности в последовательностных схемах	72
2.9. Операторы процедурного назначения	73
2.9.1. Оператор блокирующего назначения «= \Rightarrow »	73
2.9.2. Оператор неблокирующего назначения « \leq =»	74
2.9.3. Особенности синтеза операторов блокирующего и неблокирующего назначения	75
2.9.4. Использование блокирующих и неблокирующих назначений в одном процедурном блоке	77
2.10. Операторы процедурного программирования	79
2.10.1. Оператор if-else	79
2.10.2. Оператор case	82
2.10.3. Операторы casez и casex	84
2.10.4. Атрибуты full_case и parallel_case	86
2.10.5. Оператор for	91
2.10.6. Оператор while	92
2.10.7. Оператор repeat	92
2.10.8. Оператор forever	93
2.10.9. Оператор disable	94
2.11. Блок генерации	95
2.12. Задачи и функции	98
2.12.1. Автоматические и статические задачи и функции	98
2.12.2. Задачи	98
2.12.3. Функции	100
2.12.4. Константные функции	102
2.13. Системные задачи и функции	103
2.14. Директивы компилятора	104
Выводы	104
Глава 3. Введение в систему автоматизированного проектирования Quartus Prime	109
3.1. Основные этапы разработки проектов на FPGA	109
3.2. Создание нового проекта в CAD Quartus	110
3.3. Ввод кода проекта на языке Verilog	116
3.4. Использование шаблонов языка Verilog	118
3.4.1. Готовые проекты	118
3.4.2. Конструкции языка Verilog	119
3.4.3. Логические элементы	119
3.4.4. Атрибуты синтеза	120
3.4.5. Примитивы Altera	121
3.5. Добавление файлов к проекту	121
3.6. Компиляция проекта	123
3.7. Просмотр и анализ результатов синтеза	124
3.8. Ошибки компиляции	126
3.9. Назначение контактов	128
3.9.1. Назначение контактов с помощью редактора назначений	128

3.9.2. Назначение контактов с помощью файла установок	131
3.9.3. Импорт и экспорт файла установок	131
3.9.4. Использование файла установок для платы DE1-SoC	132
3.9.5. Использование имен сигналов, определенных в файле установок, в качестве внешних сигналов	133
3.10. Моделирование спроектированной схемы	135
3.10.1. Формирование тестовых векторов	135
3.10.2. Группирование и разгруппирование сигналов	138
3.10.3. Изменение порядка битов в шине	139
3.10.4. Выполнение функционального моделирования	139
3.10.5. Выполнение временного моделирования	141
3.11. Программирование и конфигурация микросхемы FPGA	141
3.11.1. Установка драйвера USB-Blaster	142
3.11.2. Программирование платы DE1-SoC	144
3.12. Тестирование спроектированной схемы	147
Выводы	147
Глава 4. Проектирование комбинационных схем	149
4.1. Возможности языка Verilog для проектирования комбинационных схем	149
4.1.1. Примитивы	149
4.1.2. Примитивы пользователя	150
4.1.3. Оператор непрерывного назначения assign	150
4.1.4. Процедурные операторы	150
4.1.5. Функции и задачи	154
4.2. Методика проектирования комбинационных схем	154
4.2.1. Описание функционирования комбинационной схемы	154
4.2.2. Минимизация булевых функций	156
4.2.3. Представление комбинационной схемы в виде совокупности логических уравнений	157
4.2.4. Представление комбинационной схемы в виде алгоритма	157
4.2.5. Написание кода комбинационной схемы на языке Verilog	158
4.3. Опасность автоматического введения защелок при синтезе комбинационных схем	158
Выводы	159
Глава 5. Проектирование стандартных функциональных узлов комбинационного типа	161
5.1. Буферы с тремя состояниями	161
5.2. Мультиплексоры	162
5.2.1. Способы описания мультиплексоров	163
5.2.2. Шинные мультиплексоры	165
5.2.3. Увеличение числа входов мультиплексора	166
5.3. Демультимплексоры	170
5.4. Шифраторы и приоритетные шифраторы	177

5.4.1. Шифраторы	177
5.4.2. Приоритетные шифраторы	184
5.5. Дешифраторы	189
5.5.1. Способы описания дешифраторов	190
5.5.2. Дешифратор с входом разрешения	195
5.5.3. Неполные дешифраторы	196
5.5.4. Дешифраторы 7-сегментных дисплеев	198
5.5.5. Дешифраторы для отображения текста на 7-сегментных дисплеях	201
5.5.6. Дешифраторы адреса	202
Выводы	202
Глава 6. Проектирование элементов последовательностных схем	205
6.1. Защелки	206
6.1.1. Защелки типа SR	206
6.1.2. Прозрачный D-триггер (защелка типа D)	210
6.1.3. Ошибочное описание D-защелки	213
6.1.4. Защелка master-slave	214
6.1.5. Ошибочная установка защелок на выходах комбинационных схем	215
6.2. Триггеры	216
6.2.1. Триггер типа D	216
6.2.2. Другие типы триггеров (SR, JK и T)	220
6.2.3. Определение начальных значений состояний триггеров	222
6.3. Регистры	223
6.4. Сдвиговые регистры	228
6.5. Циклические сдвиговые регистры	233
6.6. Сдвиговые регистры с линейной обратной связью (LFSR)	234
6.7. Регистровые файлы	235
Выводы	236
Глава 7. Счетчики, схемы задержки, делители частоты, таймеры, часы реального времени	239
7.1. Счетчики	239
7.1.1. Структурное описание счетчика	240
7.1.2. Поведенческое описание счетчика	242
7.1.3. Счетчики по модулю	243
7.1.4. Загрузка данных в счетчики	245
7.1.5. Реверсивные счетчики	249
7.1.6. Описание счетчика в виде конечного автомата	252
7.1.7. Исследование результатов реализации счетчиков на различных семействах FPGA	256
7.2. Счетчики Грея и Джонсона	258
7.2.1. Счетчик Грея	258
7.2.2. Счетчик Джонсона	259
7.3. Схемы задержек	260
7.4. Делители частоты	262

7.5. Генераторы медленных импульсов	263
7.6. Медленные счетчики	264
7.7. Таймеры	267
7.8. Часы реального времени	268
Выводы	271
Глава 8. Проектирование арифметических устройств	273
8.1. Сумматоры	273
8.1.1. Одноразрядные сумматоры: полный сумматор (full adder) и полусумматор (half adder)	274
8.1.2. Сумматор с последовательным переносом	275
8.1.3. Сумматор с параллельным (ускоренным) переносом	277
8.1.4. Сумматор с выбором переноса	286
8.1.5. Сумматор с пропуском переноса	290
8.1.6. Сумматор с условным сложением	293
8.1.7. Экспериментальные исследования реализации сумматоров на различных семействах FPGA	295
8.2. Сумматор с сохранением (запоминанием) переноса	297
8.3. Вычитатели	298
8.4. Последовательный сумматор на основе одного полного сумматора	299
8.5. Накапливающий сумматор (аккумулятор)	302
8.6. Сумматор чисел в двоично-десятичном коде	303
8.7. Инкременторы и декременторы	305
8.7.1. Инкрементор	305
8.7.2. Декрементор	309
8.8. Реализация счетчика по возрастанию с помощью инкрементора	311
Выводы	312
Глава 9. Проектирование умножителей	314
9.1. Матричный умножитель с сохранением переносов	314
9.2. Умножитель с использованием сумматоров с последовательным переносом	319
9.3. Древовидный умножитель Уоллеса	323
9.4. Умножитель Дадда	331
9.5. Алгоритмические методы реализации умножителей	335
9.6. Умножитель Бута	340
9.7. Модифицированный умножитель Бута	344
9.8. Экспериментальные исследования реализации умножителей на различных семействах FPGA	347
Выводы	349
Глава 10. Проектирование конечных автоматов	351
10.1. Структурные модели конечных автоматов	351
10.2. Представление конечных автоматов	354
10.3. Проектирование автомата Мили	356

10.3.1. Описание автомата Мили	356
10.3.2. Синтез автомата Мили	358
10.3.3. Анализ функционирования автомата Мили	360
10.4. Проектирование автомата Мура	363
10.4.1. Описание автомата Мура	363
10.4.2. Синтез автомата Мура	364
10.4.3. Анализ функционирования автомата Мура	365
10.5. Стили описания конечных автоматов	368
10.5.1. Описание конечных автоматов с двумя процессами	368
10.5.2. Описание конечных автоматов с одним процессом	371
10.5.3. Описание конечных автоматов с регистрами на выходах	374
10.5.4. Описание конечных автоматов с помощью операторов case	374
10.5.5. Описание надежных конечных автоматов	375
10.5.6. Сравнение стилей описания конечных автоматов	379
10.6. Кодирование внутренних состояний конечных автоматов	381
10.6.1. Кодирование внутренних состояний при реализации конечных автоматов на FPGA	381
10.6.2. Способы кодирования внутренних состояний конечных автоматов в системе Quartus	383
10.6.3. Кодирование внутренних состояний конечных автоматов для минимизации стоимости реализации	385
10.6.4. Сравнение способов кодирования внутренних состояний	386
10.7. Проектирование безопасных конечных автоматов	388
10.8. Использование значений входных и выходных переменных для кодирования внутренних состояний конечных автоматов	389
10.8.1. Синтез автоматов класса C	390
10.8.2. Синтез автоматов класса D	394
10.8.3. Синтез автоматов класса D ₁ без расщепления внутренних состояний	397
10.8.4. Синтез автоматов класса E	399
10.8.5. Синтез автоматов класса E ₁ без расщепления внутренних состояний	404
10.8.6. Синтез автоматов класса F	406
10.8.7. Сравнение структурных моделей конечных автоматов	410
10.9. Оценка эффективности методов проектирования конечных автоматов	413
10.9.1. Оптимизация конечных автоматов с помощью стилей описания на языке Verilog	414
10.9.2. Оптимизация конечных автоматов средствами системы Quartus	415
10.9.3. Определение наиболее эффективного подхода для оптимизации конечных автоматов	417
Выводы	419
Заключение	424
Список сокращений	425
Литература	427
Предметный указатель	428