

СОДЕРЖАНИЕ

Об авторе.....	13
Предисловие.....	14
Введение.....	16
Благодарности.....	18

1 Элементы проектирования встраиваемых систем.....	19
1.1. Уровни абстрагирования.....	20
1.1.1. От транзисторов к программам.....	20
1.1.2. Смещение уровней абстракции.....	22
1.1.3. Описание проекта.....	23
1.2. Маршрут проектирования встраиваемых систем.....	23
1.2.1. Разделение на аппаратную и программную части.....	23
1.2.2. Аппаратная часть.....	24
1.2.3. Программная часть.....	24
1.2.4. Спецификация межсоединений.....	25
1.2.5. Совместное аппаратно-программное моделирование.....	25
1.2.6. Синтез аппаратной части.....	25
1.2.7. Компиляция программной части.....	26
1.2.8. Генерация межсоединений.....	27
1.2.9. Интеграция проекта.....	27
1.3. Средства проектирования.....	27
1.3.1. Описание структурной схемы.....	28
1.3.2. Языки описания аппаратуры и другие программы моделирования аппаратной части.....	28
1.3.3. Компиляторы языков программирования.....	28
1.3.4. Программа моделирования системы, описанной в виде списка цепей.....	28
1.3.5. Эмулятор системы команд.....	28
1.3.6. Программы синтеза аппаратной части.....	29
1.3.7. Компилятор в машинные коды.....	29
1.3.8. Программы сборки и отладки программной части.....	29
1.3.9. Программа-интегратор.....	29
1.4. Новые тенденции проектирования аппаратуры.....	29
1.4.1. Конфигурируемые процессоры.....	29
1.4.2. Стандартные шины.....	30
1.4.3. Программирование.....	30
1.4.4. Программное обеспечение.....	30
1.5. Выводы.....	30

2 Основы проектирования логических схем.....	31
2.1. Системы счисления.....	32
2.1.1. Двоичные числа.....	33

2.1.2. Шестнадцатеричные числа.....	33
2.2. Двоичная арифметика.....	34
2.2.1. Числа со знаком.....	34
2.2.2. Двоичное сложение.....	35
2.2.3. Двоичное вычитание.....	35
2.2.4. Система дополнения до двух.....	35
2.2.5. Переполнение.....	36
2.2.6. Числа с фиксированной запятой.....	37
2.2.7. Числа с плавающей запятой.....	37
2.3. Базовые логические вентили и структуры.....	39
2.3.1. Система логических значений.....	39
2.3.2. Представление логических функций.....	39
2.3.3. Транзисторы.....	40
2.3.4. КМОП-инвертор.....	41
2.3.5. КМОП-вентиль И-НЕ.....	41
2.3.6. КМОП-вентиль ИЛИ-НЕ.....	42
2.3.7. Вентили И и ИЛИ.....	43
2.3.8. Вентиль ИСКЛЮЧАЮЩЕЕ ИЛИ.....	43
2.3.9. Мультиплексоры.....	44
2.3.10. Вентили с тремя состояниями выходов.....	45
2.3.11. Функциональный генератор (LUT-элемент).....	46
2.4. Проектирование комбинационных схем.....	47
2.4.1. Булева алгебра.....	47
2.4.2. Карты Карно.....	49
2.4.3. Неопределенные значения.....	53
2.4.4. Минимальное покрытие.....	54
2.4.5. Итеративная аппаратура.....	55
2.4.6. Мультиплексоры и дешифраторы.....	58
2.4.7. Уровни активности.....	60
2.4.8. Входы разрешения/запрета.....	61
2.4.9. Высокоуровневое проектирование.....	61
2.5. Запоминающие элементы.....	62
2.5.1. Простой триггер-защелка.....	63
2.5.2. Тактируемый D-триггер.....	64
2.5.3. D-триггер (триггер с задержкой).....	65
2.5.4. Управление триггером.....	66
2.5.5. Регистры.....	68
2.6. Проектирование последовательностных схем.....	68
2.6.1. Конечные автоматы.....	68
2.6.2. Проектирование конечных автоматов.....	70
2.6.3. Автоматы Мили и Мура.....	75
2.6.4. Унарное кодирование состояний.....	76
2.6.5. Часто используемые последовательностные схемы.....	77
2.7. Запоминающие устройства.....	80
2.7.1. Структура статического оперативного запоминающего устройства.....	81

2.8. Двухнаправленные выводы.....	82
2.9. Обобщающий пример: последовательный сумматор.....	82
2.9.1. Постановка задачи.....	83
2.9.2. Разбиение проекта на блоки.....	83
2.9.3. Проектирование операционного устройства.....	84
2.10. Выводы.....	87

3 Проектирование на уровне регистровых передач с использованием языка Verilog.....	89
3.1. Основные структуры языка Verilog.....	90
3.1.1. Модули.....	92
3.1.2. Структура модуля.....	93
3.1.3. Порты модуля.....	94
3.1.4. Переменные модуля.....	95
3.1.5. Система логических значений.....	97
3.1.6. Вычисление значений проводных соединений (цепей).....	97
3.2. Комбинационные схемы.....	99
3.2.1. Комбинационные схемы на уровне вентилях.....	99
3.2.2. Синтез на вентилях.....	104
3.2.3. Описания с использованием логических уравнений.....	105
3.2.4. Создание других модулей.....	109
3.2.5. Синтез операторов присваивания.....	111
3.2.6. Описания с использованием процедурных операторов.....	112
3.2.7. Правила описания комбинационных схем.....	117
3.2.8. Синтез процедурных блоков.....	117
3.2.9. Соединения с шиной.....	119
3.3. Последовательностные схемы.....	119
3.3.1. Основные элементы памяти на вентилях.....	120
3.3.2. Элементы памяти, использующие процедурные операторы.....	121
3.3.3. Синтез триггеров.....	125
3.3.4. Регистры, сдвиговые регистры и счетчики.....	127
3.3.5. Синтез сдвиговых регистров и счетчиков.....	130
3.3.6. Кодирование конечных автоматов.....	130
3.3.7. Синтез конечных автоматов.....	140
3.3.8. Запоминающие устройства.....	142
3.4. Написание тестовых примеров.....	144
3.4.1. Генерация периодических данных.....	145
3.4.2. Случайные входные данные.....	146
3.4.3. Привязка данных ко времени.....	147
3.5. Спецификация последовательного умножителя.....	148
3.5.1. Процесс умножения путем сдвига и сложения.....	148
3.5.2. Проектирование последовательного умножителя.....	151
3.5.3. Тестирование умножителя.....	156
3.6. Синтез результатов.....	160
3.7. Выводы.....	160

4	Аппаратное и программное обеспечение компьютера	162
4.1.	Аппаратное и программное обеспечение компьютера.....	163
4.2.	Программное обеспечение компьютера.....	166
4.2.1.	Машинный код.....	166
4.2.2.	Язык ассемблера.....	166
4.2.3.	Язык высокого уровня.....	166
4.2.4.	Язык программирования С.....	168
4.3.	Архитектура множества команд	179
4.3.1.	Аппаратное обеспечение	179
4.3.2.	Программное обеспечение	180
4.3.3.	Аппаратно-программный интерфейс	180
4.4.	Проектирование процессора SMPL-CPU	181
4.4.1.	Описание процессора	181
4.4.2.	Одноцикловая реализация	183
4.4.3.	Многоцикловая реализация.....	196
4.5.	Проектирование и тестирование компьютера SAYEH.....	207
4.5.1.	Детали функционирования процессора.....	207
4.5.2.	Операционное устройство процессора SAYEH.....	210
4.5.3.	Описание процессора SAYEH на языке Verilog.....	212
4.5.4.	Тестовый пример/ассемблер верхнего уровня процессора SAYEH	213
4.5.5.	Реализация аппаратной части процессора SAYEH.....	215
4.6.	Выводы	215
5	Программируемые логические интегральные схемы (ПЛИС)	216
5.1.	Постоянные запоминающие устройства.....	217
5.1.1.	Основная структура ПЗУ	217
5.1.2.	Реализация на вентилях ИЛИ-НЕ	219
5.1.3.	Распределенные вентили.....	219
5.1.4.	Программируемость матриц	221
5.1.5.	Обзор памяти.....	221
5.1.6.	Разновидности ПЗУ.....	222
5.2.	Программируемые логические матрицы	226
5.2.1.	Логическая структура PAL	228
5.2.2.	Расширение промежуточных шин	229
5.2.3.	Выходы с тремя состояниями.....	230
5.2.4.	Регистровые выходы.....	231
5.2.5.	Коммерческие устройства.....	231
5.3.	Сложные программируемые логические устройства.....	234
5.3.1.	CPLD семейства MAX 7000S фирмы Altera.....	235
5.4.	Программируемые пользователем вентиляльные матрицы.....	237
5.4.1.	FPGA семейства FLEX 10K фирмы Altera	238
5.4.2.	FPGA семейства Cyclone фирмы Altera.....	245
5.5.	Выводы	264

6 Средства для проектирования и макетирования	265
6.1. Маршрут проектирования аппаратной части	266
6.1.1. Операционное устройство последовательного сумматора.....	267
6.1.2. Контроллер последовательного сумматора	268
6.2. HDL-моделирование и HDL-синтез	270
6.2.1. Моделирование перед синтезом	271
6.2.2. Синтез модуля.....	277
6.2.3. Моделирование после выполнения синтеза.....	281
6.3. Смешанно-уровневое проектирование в системе QuartusII	284
6.3.1. Описание проекта	286
6.3.2. Файл структурной схемы проекта	287
6.3.3. Создание и вставка элементов проекта	288
6.3.4. Элементы проводных соединений проекта.....	294
6.3.5. Компиляция проекта	295
6.3.6. Моделирование проекта	296
6.3.7. Результаты синтеза.....	299
6.4. Макетирование проекта.....	304
6.4.1. Описание платы UP3.....	304
6.4.2. Спецификация платы DE2	311
6.4.3. Программирование устройства Cyclone II на плате DE2	320
6.5. Выводы	324

7 Проектирование аппаратных утилитных ядер	325
7.1. Управление библиотекой	326
7.2. Руководство по основным устройствам ввода-вывода.....	327
7.2.1. Устройство устранения дребезга.....	327
7.2.2. Устройство формирования одного импульса	331
7.2.3. Использование основных элементов ввода-вывода платы UP3	333
7.2.4. Использование основных элементов ввода-вывода платы DE2	335
7.3. Делители частоты	336
7.4. Семисегментные дисплеи	336
7.4.1. Дешифратор для семисегментного дисплея.....	337
7.4.2. Тестирование дешифратора для сегментного дисплея платы DE2.....	337
7.5. Адаптер жидкокристаллического дисплея	337
7.5.1. Запись в жидкокристаллический дисплей.....	338
7.5.2. Инициализация жидкокристаллического дисплея	340
7.5.3. Драйвер дисплея с инициализацией	342
7.5.4. Тестирование драйвера жидкокристаллического дисплея на плате UP3	342
7.5.5. Тестирование драйвера жидкокристаллического дисплея на плате DE2	343
7.6. Логика интерфейса клавиатуры	344
7.6.1. Передача последовательных данных.....	344

7.6.2. Алгоритм включения напряжения питания.....	346
7.6.3. Коды и команды.....	347
7.6.4. Проектирование интерфейса клавиатуры.....	350
7.7. Логический интерфейс VGA.....	355
7.7.1. Функционирование драйвера VGA	355
7.7.2. Аппаратура синхронизации монитора.....	358
7.7.3. Дисплей символов	359
7.7.4. Драйвер VGA для текстовых данных	363
7.7.5. Макетирование драйвера VGA на плате UP3	364
7.7.6. Макетирование драйвера VGA на плате DE2	364
7.8. Выводы	366
<hr/>	
8 Проектирование со встраиваемыми процессорами.....	367
8.1. Этапы встраиваемого проектирования	368
8.1.1. Выбор процессора	368
8.1.2. Сопряжение процессора	371
8.1.3. Разработка программного обеспечения	372
8.2. Проектирование фильтра.....	372
8.2.1. Концепции фильтров.....	372
8.2.2. Аппаратная реализация КИХ-фильтра.....	376
8.2.3. Встраиваемая реализация КИХ-фильтра.....	378
8.2.4. Построение КИХ-фильтра.....	383
8.3. Проектирование микроконтроллера.....	385
8.3.1. Системная платформа	385
8.3.2. Архитектура микроконтроллера.....	386
8.4. Выводы	387
<hr/>	
9 Проектирование встраиваемой системы.....	388
9.1. Проектирование встраиваемой системы	389
9.2. Процессор Nios II.....	390
9.2.1. Свойства конфигурируемости процессора Nios II	392
9.2.2. Архитектура процессора.....	394
9.2.3. Система команд	400
9.2.4. Альтернативные ядра процессора Nios II	403
9.3. Переключательная структура Avalon.....	403
9.3.1. Спецификация Avalon	403
9.3.2. Логика дешифрации адреса.....	406
9.3.3. Мультиплексирование маршрутов данных	406
9.3.4. Вставка состояния ожидания.....	407
9.3.5. Конвейерная обработка	407
9.3.6. Преобразование порядка байтов	408
9.3.7. Собственное выравнивание адреса и динамический размер шины	408
9.3.8. Арбитраж для многомастерных систем.....	409

9.3.9. Управление пакетами.....	411
9.3.10. Пересечение областей синхронизации.....	412
9.3.11. Контроллер прерываний.....	412
9.3.12. Распределение сигнала сброса.....	413
9.4. Обзор программы SOPC Builder.....	414
9.4.1. Архитектура систем, создаваемых программой SOPC Builder.....	414
9.4.2. Функции программы SOPC Builder.....	416
9.5. Интегрированная среда проектирования IDE.....	417
9.5.1. Менеджер проекта IDE.....	417
9.5.2. Редактор исходного кода.....	418
9.5.3. Компилятор языка C/C++.....	418
9.5.4. Отладчик.....	418
9.5.5. Программатор flash-памяти.....	419
9.6. Проект встраиваемой системы: калькулятор.....	419
9.6.1. Описание системы.....	420
9.6.2. Вычислительное устройство.....	420
9.6.3. Интерфейс ввода-вывода калькулятора.....	421
9.6.4. Проектирование вычислительного устройства.....	422
9.6.5. Построение программного обеспечения калькулятора.....	430
9.6.6. Программа калькулятора.....	435
9.6.7. Завершение проекта системы «калькулятор».....	439
9.7. Выводы.....	441
<hr/>	
Приложение А. Множество команд процессора Nios II.....	442
А.1. Команды передачи данных.....	442
А.2. Арифметические и логические команды.....	443
А.3. Команды пересылки.....	443
А.4. Команды сравнения.....	444
А.5. Команды сдвига и ротации.....	444
А.6. Команды управления программой.....	445
А.7. Другие команды управления.....	445
А.8. Пользовательские команды.....	446
А.9. Команда <i>нет операции</i>	446
А.10. Потенциально нереализованные команды.....	447
<hr/>	
Приложение В. Литература для дополнительного чтения.....	448
Предметный указатель.....	450

ОБ АВТОРЕ

Доктор Заиналабедин Наваби является профессором электротехники и вычислительной техники Северо-восточного университета. Доктор Наваби является автором нескольких учебников и электронных учебных пособий по языкам проектирования VHDL и Verilog, а также соответствующих средств и программных пакетов. Доктор Наваби был вовлечен в проблему языков описания аппаратуры в 1976 году, когда он начал разрабатывать программу моделирования на уровне регистровых передач (Register-Transfer Level – RTL) для одного из первых языков описания аппаратуры (Hardware Description Languages – HDLs). В 1981 году он завершил разработку средства синтеза, которое генерировало рисунок МОП-структуры (МОП – металл-оксид-полупроводник – Metal-Oxide-Silicon – MOS) из HDL-описания. Начиная с 1981 года, доктор Наваби был вовлечен в проектирование, определение спецификаций и реализацию языков описания аппаратуры. Ему принадлежат многочисленные статьи по применению языков описания аппаратуры при моделировании, синтезе и тестировании цифровых систем. В 1990 году он был одним из первых, кто начал читать полные курсы лекций по языкам описания аппаратуры в Северо-восточном университете. С тех пор он провел много кратких курсов и консультаций по данной тематике в США и за границей. В дополнение к должности профессора он также является консультантом компаний по автоматизации проектирования электроники (Electronic Design Automation – EDA). Доктор Наваби получил степень магистра и доктора философии (Ph. D. – Philosophy Doctor) от Университета в Аризоне в 1978 и 1981 годах соответственно, а также степень бакалавра от Техасского университета города Остина (США) в 1975 году. Он является старшим членом общества IEEE (The Institute of Electrical and Electronics Engineers – Институт инженеров по электротехнике и радиоэлектронике), а также членом следующих обществ: IEEE по вычислительной технике, ASEE и ACM (Association for Computing Machinery – Ассоциация по вычислительной технике). Доктор Наваби является автором шести книг по различным аспектам автоматизации проектирования цифровых систем.

ПРЕДИСЛОВИЕ

Темой данной книги является проектирование встраиваемых систем на программируемых логических интегральных схемах (ПЛИС). Проектирование встраиваемых систем включает в себя проектирование функций, которые могут быть реализованы в аппаратной части системы или могут реализовываться как выполнение программы на встраиваемом процессоре. Природа встраиваемых систем базируется на нескольких дисциплинах. Поэтому проектирование встраиваемых систем включает концепции цифровых систем, архитектур компьютеров, разработку программного обеспечения, а также проектирование компьютерных систем и систем на базе микропроцессоров. Кроме того, реализация встраиваемых систем на ПЛИС требует дополнительного знания программируемых устройств, а также соответствующих средств и языков проектирования (например, VHDL или Verilog).

Очевидно, что полное освещение всех этих тем в одной книге невозможно. С другой стороны, нет необходимости углубляться в каждую из этих тем для того, чтобы стать разработчиком встраиваемых систем. Разработчик цифровой системы смотрит на проектирование цифровой системы с точки зрения системы, при этом ему требуется знание основ всех упомянутых предметов.

Книга «Проектирование встраиваемых систем на базе ПЛИС» предоставляет всю информацию, которая необходима для проектирования сложных встраиваемых систем и ядер. Первая часть определяет, чем является встраиваемая система и знание каких аспектов цифрового проектирования полезно при проектировании таких систем. Затем мы совместим логическое проектирование с современным проектированием на уровне регистровых передач (Register Transfer Level – RTL). Потом обсудим язык проектирования Verilog на уровне регистровых передач и покажем применение этого языка для RTL-моделирования¹ и RTL-синтеза². После обсуждения аппаратной части на уровне регистровых передач и соответствующих методологий проектирования мы перейдем к аппаратной и программной частям компьютеров. Поскольку материал уровня регистровых передач является полезным для аппаратной части проектов встраиваемых систем, материал компьютерных систем и их архитектур предназначается для программной части таких проектов. После рассмотрения основ мы покажем, как эти темы совместить вместе в проекте законченной системы с аппаратными и программными ядрами. На всем протяжении этой презентации вводятся и используются современные средства проектирования встраиваемых систем на уровне регистровых передач.

Эта книга может использоваться профессиональными разработчиками аппаратуры, кто уже знаком с основами логического проектирования и желает заняться автоматизированным проектированием на системном уровне. Для этой аудитории книга включает разделы проектирования цифровых систем, архитектуры компьютеров и разработку программной части системы. Она также содер-

¹ RTL-моделирование – моделирование на уровне регистровых передач. – *Прим. перев.*

² RTL-синтез – синтез на уровне регистровых передач. – *Прим. перев.*

жит программы, показывающие использование языка проектирования Verilog, а также компиляторы и ассемблеры для проектов встраиваемых систем. Дополнительно для индустриального применения мы покажем, как существующие аппаратные и программные проектные компоненты, а также библиотечные ядра используются на верхнем уровне проектов. Используя отладочные платы фирмы Altera, эта книга дает руководящие знания по охватываемым темам.

В учебном процессе книга может использоваться на верхнем уровне факультативных технических курсов для студентов специальностей вычислительная техника и электротехника, а также других инженерных специальностей. Совместно с отладочными платами фирмы Altera эта книга помогает студентам видеть свои проекты в процессе разработки: как проекты реализуются и тестируются, вследствие чего студенты получают понимание, как данные вещи работают. Для студентов других инженерных направлений, например механической или химической, книга является полезным средством для проектирования и реализации контроллеров и интерфейсов.

Заиналабедин Наваби
navabi@ece.neu.edu
Бостон, Массачусетс, США
Июль, 2006

ВВЕДЕНИЕ

Проектирование встраиваемых систем, которое стало новой тенденцией в создании аппаратуры, использует встраиваемые ядра и процессоры как компоненты цифровых систем. Разработчик встраиваемой системы для представления различных частей своего проекта использует соединение высокоуровневого программного обеспечения и описания аппаратной части на уровне регистровых передач. Разработчик аппаратуры должен сознавать, что абстрактные связи между различными частями системы также становятся результатом проектирования. Кроме того, важно, чтобы разработчик знал обо всех средствах, которые доступны для проектирования и реализации аппаратуры.

Ранняя практика проектирования цифровых систем на основе отдельных транзисторов сменилась проектированием на вентиляльном уровне, а в конце 80-х годов доминирующей технологией становится проектирование на уровне регистровых передач. Сегодня мы видим, что уровень регистровых передач является слишком детальным, поэтому требуется более высокий уровень абстрагирования. Уровень электронной системы (Electronic System Level – ESL) и является этим следующим уровнем абстрагирования. На этом уровне разработчики описывают свои аппаратные компоненты и с помощью средств проектирования транслируют свои описания на более детальный уровень регистровых передач или уровень вентилялей.

По мере усложнения проектов и повышения уровня описания аппаратуры соединения между ее компонентами становятся более изоциренными. В то время как простые проводные связи используются для соединений между транзисторами, логические сигналы (с большим функциональным значением) становятся межсоединениями для проектов на вентиляльном уровне. Межсоединения становятся более сложными (превращаясь в шины), когда мы переходим от вентиляльного уровня к уровню регистровых передач. Продолжая эту тенденцию, сегодня мы можем видеть, что межсоединения системного уровня становятся даже более сложными и сами содержат сложные RTL-компоненты. Простые шинные межсоединения уровня регистровых передач стали интеллектуальными системами шин или переключательными структурами, которые управляют передачами блоков данных, арбитражем и различными формами коммуникаций «главный–подчиненный» (master–slave). Понимание этой функции межсоединений является критически важным для сегодняшних разработчиков системного уровня.

Сегодняшним разработчикам аппаратуры требуются умение создавать программы для реализации аппаратных функций; понимание архитектур процессоров, на которых выполняются эти программы; знание основ проектирования аппаратуры на уровне регистровых передач, а также навыки описания межсоединений системных компонент.

Данная книга освещает уровень регистровых передач, методологию проектирования на системном уровне, программируемые логические интегральные схемы (ПЛИС), а также средства и интегрированные среды, которые имеются в распоряжении разработчика системного уровня. Книга может использоваться в академических или индустриальных средах студентами или инженерами.

В обоих случаях требуется знание общих принципов логического проектирования. После рассмотрения этих тем в книге описаны способы проектирования на уровне логических вентилей для охвата уровня регистровых передач. Первые пять глав охватывают главные концепции цифрового проектирования на ПЛИС с практической точки зрения. Оставшиеся главы показывают интегрированные среды для проектирования ядер и реализации проектов на системном уровне, используя аппаратные и программные ядра.

В главе 1 обсуждаются общий маршрут проектирования системного уровня и роль компиляторов и средств синтеза. Показывается, что необходимо для того, чтобы стать инженером-проектировщиком встраиваемых систем.

В главе 2 с практической точки зрения обсуждается логическое проектирование уровня регистровых передач. Рассматриваемые здесь темы в основном используются при автоматизированном проектировании на уровне регистровых передач.

В главе 3 вводится язык проектирования Verilog. Особо подчеркиваются возможности языка Verilog при проектировании на уровне регистровых передач. Для завершения проекта обсуждаются вопросы построения тестовых примеров и использования языковых утилит.

В главе 4 обсуждаются компьютерные системы, компьютерные архитектуры и высокоуровневый язык проектирования C. В этой главе показывается, как взаимодействуют аппаратная и программная части процессора.

В главе 5 детально рассматриваются программируемые логические устройства. Показывается, как оригинальные ПЗУ включаются в современные сложные ПЛИС.

В главе 6 обсуждаются средства, которые мы используем для утверждения спецификации проекта, его синтеза, программирования ПЛИС и макетирования всего проекта. Мы обсуждаем использование пакета Quartus II, программы моделирования ModelSim HDL и отладочных плат DE2 и UP3.

В главе 7 показывается несколько интерфейсных проектов. В этих проектах показано, как ядра создаются и используются.

В главе 8 показаны элементы законченного проекта встраиваемой системы, который включает аппаратную и программную части. Мы покажем встраиваемую реализацию FIR-фильтра. Для того чтобы продемонстрировать детали, мы не будем использовать преимущества средств проектирования и интегрированных пакетов, которые доступны для проектов встраиваемых систем.

В главе 9 показано, как проектируются и реализуются компоненты сложной системы, которая имеет аппаратную и программную части. Для этого проекта мы покажем использование всех возможностей проектирования, программные средства и средства автоматизации проектирования, которые имеются в распоряжении у разработчика встраиваемой системы.

1 Элементы проектирования встраиваемых систем

2	Основы проектирования логических систем	31
3	Проектирование на уровне регистровых передач с использованием языка Verilog	89
4	Аппаратное и программное обеспечение компьютера	162
5	Программируемые логические интегральные схемы (ПЛИС)	216
6	Средства для проектирования и макетирования	265
7	Проектирование аппаратных утилитных ядер	325
8	Проектирование со встраиваемыми процессорами	367
9	Проектирование встраиваемой системы	388

Встраиваемая система является цифровой системой с одним или несколькими процессорами, которые реализуют ее аппаратные функции. Процессоры встраиваемой системы называются *встраиваемыми процессорами*. Встраиваемые системы облегчают проектирование *цифровых систем*, давая разработчикам возможность использовать программы на языках С или С++ для описания и проектирования сложных аппаратных функций. Программа на языке высокого уровня заменяет детальное проектирование аппаратуры, которое обычно осуществлялось написанием синтезируемого *HDL-кода* или использованием аппаратных библиотечных компонент.

Отметим, что проектирование встраиваемой системы, определенной выше, не сильно отличается от проектирования *микроконтроллеров*. Встраиваемые процессоры используются для реализации аппаратуры точно таким же образом, как микроконтроллеры. Главное отличие заключается в том, что встраиваемые системы предоставляют больше гибкости и могут реализовывать более сложные заказные проекты. Отдельные *аппаратные компоненты* (части) встраиваемой системы также могут реализовываться на встраиваемых процессорах. *Методология проектирования встраиваемых систем* предоставляет собой методы высокоуровневого проектирования для объединения частей системы, реализуемых на встраиваемом процессоре.

Для того чтобы было возможным получить пользу от гибкости и преимуществ высокоуровневого проектирования, предоставляемых встраиваемыми системами, должна быть изучена новая методология проектирования аппаратуры. Эта методология включает использование аппаратуры и программного обеспечения в одном и том же интегрированном проектном окружении (интегрированном пакете). В данной главе особое внимание уделяется элементам встраиваемого проектирования. Мы обсудим методологию проектирования встраиваемых систем, роль программного обеспечения, роль *языков описания аппаратуры* (Hardware Description Languages – HDLs), объединение аппаратных и программных частей, а также средства и интегрированные программные пакеты, которые доступны для проектирования встраиваемых систем.

1.1. Уровни абстрагирования

Проектирование цифровых систем развивалось от уровня транзисторов к вентильному уровню и затем – к уровню регистровых передач (Registered Transfer Level – RTL). Хотя языки описания аппаратуры и библиотеки конфигурируемых компонент предоставляют разработчику RTL-уровня быстрые способы создания проектов, для реализации современных сложных аппаратных систем необходимы более высокие уровни абстракции.

1.1.1. От транзисторов к программам

Ранее проектирование цифровых устройств начиналось с размещения транзисторов для реализации нужной функции. Очевидно, что при таком ручном методе проектирования гибкость заключается в выборе размеров транзисторов и спосо-

бе проведения проводных соединений, однако при этом достигается оптимальная реализация заданной функции.

Но, когда проекты становятся более сложными, сам метод проектирования должен измениться, для того чтобы стало возможным создание больших схем. В процессе эволюции проектирование на уровне транзисторов преобразовалось в проектирование на уровне вентилях. В процессе перехода к более высокому уровню абстракции пришлось искать компромиссы между быстродействием, использованием площади кристалла (то есть стоимостью) и потребляемой мощностью. Кроме того, средства проектирования должны были обеспечивать верификацию проекта на уровне вентилях и его трансляцию на уровень транзисторов.

По мере усложнения проектов возникла необходимость в еще более высоком уровне абстракции, при котором количество элементов уменьшилось бы, по сравнению с числом вентилях. Основное внимание на этом уровне абстракции сосредоточено на передаче данных между регистрами, логическими узлами и шинами. Поэтому он и называется уровнем регистровых передач (RTL). Как и при переходе от транзисторного уровня к вентильному, переход от вентилях уровня к уровню регистровых передач сопровождается компромиссными проектными решениями. Кроме того, этот достаточно высокий уровень абстракции требует использования разнообразных программных и аппаратных средств проектирования для помощи разработчикам цифровых систем. Как и на вентильном уровне, средства проектирования уровня регистровых передач включают средства для сбора данных проекта, верификации проекта и перехода от уровня регистровых передач к низшему абстрактному уровню, то есть к вентильному уровню синтеза.

С течением времени проекты еще более усложнились, и возникла необходимость в переходе на более высокий, по сравнению с регистровыми передачами, уровень абстракции. В настоящее время этот уровень, называемый *уровнем электронной системы* (Electronic System Level – ESL), или системным уровнем, является наивысшим. На системном уровне разработчик заботится только о функционировании разрабатываемой системы и описывает алгоритм, который должен быть реализован. Алгоритм описывается с помощью процедурного языка, подобного языку программирования С. Описание системы на этом уровне не содержит синхросигналов или временных задержек вентилях уровня.

Средства проектирования системного уровня включают средства ввода, моделирования и, конечно, программы генерации аппаратной части. Генерация аппаратуры из описания системного уровня может выполняться одним из двух возможных способов. Первый способ аналогичен применяемому на других уровнях абстрагирования и заключается в трансляции описания системного уровня на низший уровень абстракции, то есть на уровень регистровых передач. Альтернативным способом является путь, когда процедурное описание системного уровня может компилироваться для выполнения на заданном процессоре. Этот метод становится возможным только на системном уровне, потому что описание системного уровня является процедурным, и для этого используется язык описания программной части системы, подобный языку С.

Именно последний вышеупомянутый метод генерации аппаратуры из описания системного уровня и должен стать методом проектирования встраиваемых систем. Традиционную методику, то есть трансляцию описания на системном

уровне в описание на уровне регистровых передач, часто называют C-синтезом, или синтезом системного уровня. На рис. 1.1 приведены обсуждаемые здесь уровни абстракции описания цифровой системы.

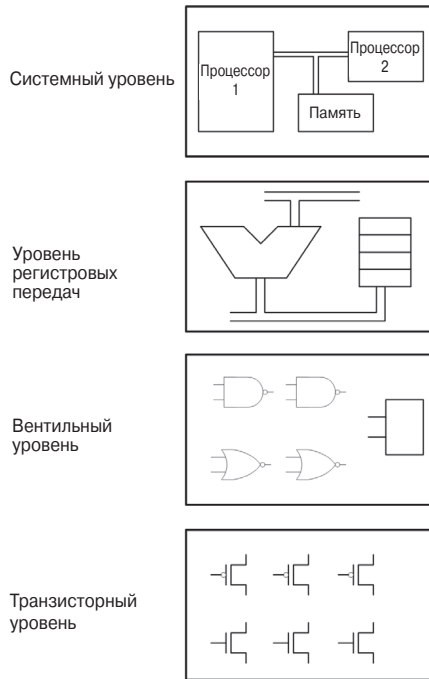


Рис. 1.1. Уровни абстракции описания цифровой системы

1.1.2. Смещение уровней абстракции

Хотя проектирование на высшем уровне абстракции проще, чем на низшем уровне, и для первого существует более развитый инструментарий, разработчики всегда используют смешение различных уровней. Выбор того или иного уровня определяется требованиями к разработке и тем, сколько труда инженер готов вложить в проектирование.

Разработчик переходит от уровня вентилях к уровню транзисторов для реализации специальных логических функций, или когда требуется оптимизация проектируемого узла. Аналогично очень часто у разработчика, действующего на уровне регистровых передач, возникает необходимость введения нескольких вентилях для реализации специальных вариантов регистровых пересылок или согласования отдельных узлов с помощью связующих логических схем (*glue logic*).

Ситуация с системным уровнем не отличается от рассмотренной выше. Нередко часть проекта системного уровня описывается на процедурном языке и реализуется с помощью встраиваемого процессора, а другая часть того же проекта описывается на уровне регистровых передач. Компоненты уровня регистровых передач могут быть реализованы более эффективно, чем на базе встраиваемого

процессора, кроме того, они оптимальны с точки зрения используемой площади кристалла, быстродействия и потребляемой мощности.

Разработчик аппаратуры системного уровня должен иметь возможность использовать средства и методы проектирования как системного уровня, так и уровня регистровых передач.

1.1.3. Описание проекта

В зависимости от уровня абстракции проект может описываться различными способами. На транзисторном и вентиляльном уровнях основным способом описания является принципиальная схема, поэтому большинство средств проектирования ориентировано именно на эту форму спецификации проекта. На уровне регистровых передач недвусмысленную и компактную форму представления проекта обеспечивают языки описания аппаратуры. Несмотря на это, для представления высокоуровневых соединений между компонентами все еще используются и структурные схемы.

На системном уровне наиболее распространенным форматом спецификации системы является программа на языке C/C++. В то же время для спецификации соединений компонент широко используется и графическое представление в виде структурной схемы. При этом отдельные компоненты структурной схемы могут описываться на системном уровне, уровне регистровых передач или, в свою очередь, являться структурными схемами.

1.2. Маршрут проектирования встраиваемых систем

На рис. 1.2 показан маршрут проектирования встраиваемой системы. По рисунку видно, что при проектировании часть функций реализуется программно, часть – аппаратно, а затем результаты объединяются в одно аппаратное исполнение. В приводимых ниже подразделах описываются детали данной структурной схемы.

1.2.1. Разделение на аппаратную и программную части

Первым этапом в проектировании встраиваемой системы является принятие решения о том, какая часть проекта будет реализована аппаратно, будучи описана в виде аппаратных блоков, HDL-программ или вентиляльных структур, а какая часть проекта будет реализована с помощью программ, выполняемых процессором. Это решение относится к задаче разделения системы на аппаратную и программную части. Такое разделение выполняется вручную (или почти вручную) и, возможно, является наиболее трудной фазой проектирования системы.

Аппаратная часть становится описанием различных аппаратных модулей, которые представляются на HDL-языках или доступны как предопределенные аппаратные модули. Программная часть является высокоуровневой программой на языке C/C++, которая после компиляции становится содержимым памяти процессора, выполняющего данную программу.

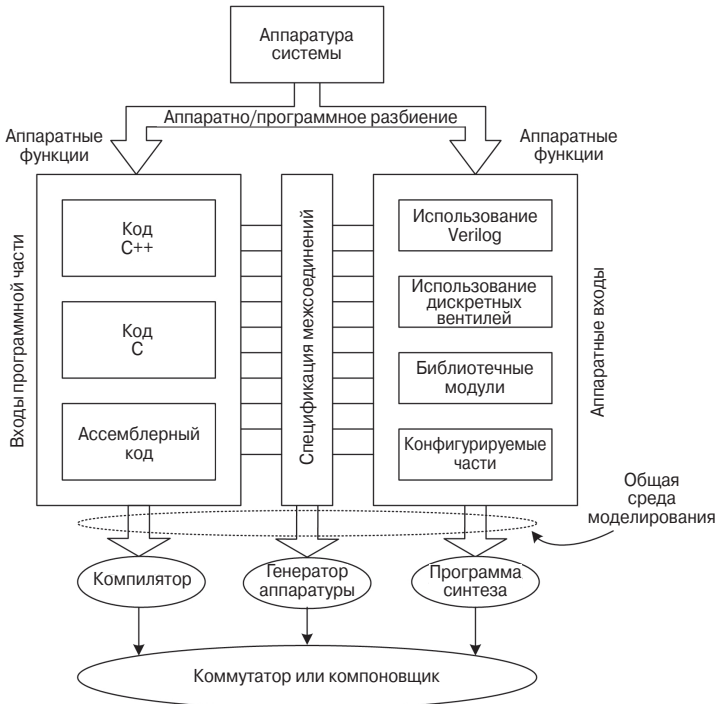


Рис. 1.2. Маршрут проектирования встраиваемых систем

1.2.2. Аппаратная часть

Аппаратная часть (правый маршрут на рис. 1.2) законченной аппаратно-программной системы может быть композицией компонентов, описанных на языке проектирования Verilog, библиотечных элементов или блоков, спроектированных на вентиляльном уровне. Используя средства и пакеты проектирования, разработчик аппаратуры выбирает, описывать части проекта на языке Verilog или использовать элементы из библиотеки predetermined модулей.

Часто средства проектирования предоставляют в распоряжение разработчика IP-блоки (IP – Intellectual Property), которые разработчики могут использовать и объединять в своих проектах для выполнения сложных задач, а также настраиваемые блоки, позволяющие реализовать часто используемые узлы типа АЛУ, регистровых банков или счетчиков.

1.2.3. Программная часть

Левый маршрут на рис. 1.2 показывает реализацию программной части системы. Часть проекта, которая реализуется программно, в конце концов должна стать программой на машинном языке выбранного процессора. При этом разработчик может выбрать либо описание этой части на языке высокого уровня с последующей компиляцией, либо программирование непосредственно на языке ассемблера или в машинных кодах.

Разработчику, использующему поддерживаемое процессорное ядро, доступны все необходимые инструменты программирования. В этом случае наилучшим выбором для описания программной части системы является использование языка C/C++, а разработчик может использовать средства компиляции и отладки. С другой стороны, если разработчик использует свой собственный процессор или процессорное ядро, которое не поддерживается в используемой им системе проектирования, то именно разработчик отвечает за генерацию программы в машинных кодах, используемых его процессором.

Независимо от того, как ведется программирование, после его окончания программная часть проекта рассматривается подобно любому аппаратному блоку, имеющему входы и выходы. Эти входы и выходы для проектируемой системы могут быть либо внешними, либо внутренними, используемыми для соединения аппаратной и программной частей.

1.2.4. Спецификация межсоединений

Средняя диаграмма на рис. 1.2 показывает блок, объединяющий программную и аппаратную части. Этот блок может быть реализован в виде простой общей шины, нескольких проводных соединений или сложной переключательной структуры. Обычно интегрированные пакеты проектирования встраиваемых систем предлагают свои собственные шинные структуры. В таких шинах предусматриваются квитирование установления связи, временные задержки, аппаратура блочной пересылки и другие высокоуровневые операции. При этом все тонкости и детали реализации шины остаются «прозрачными», невидимыми для разработчика верхнего уровня системы.

1.2.5. Совместное аппаратно-программное моделирование

Система, которая является частично аппаратной, частично программной и включает процессы обмена и передачи данных, должна моделироваться в едином интегрированном пакете для верификации логики и временных параметров проекта.

До перехода к описанию проекта на уровне вентилях и списка цепей аппаратная его часть моделируется на уровне регистровых передач, а программная – на уровне команд. Программы, использующие описание проекта на языке описания аппаратуры (HDL), обеспечивают моделирование на уровне регистровых передач, а программы-отладчики (Instruction Set Simulators – ISS) обеспечивают моделирование на уровне команд. Обычно интегрированный пакет проектирования встраиваемых систем предоставляет средства для совместного моделирования (co-simulation) всех частей законченной программно-аппаратной системы.

1.2.6. Синтез аппаратной части

Часть системы, которая специфицируется с использованием методов описания аппаратуры (правая сторона на рис. 1.2), синтезируется для получения списка цепей вентилях и базовых логических блоков. Как показано на рис. 1.3, в дополнение к описанию аппаратуры программа синтеза требует задания целевых спе-

цификаций. Целевые спецификации сообщают средству синтеза, какие ресурсы в терминах вентилях и логических блоков доступны для реализации нашей аппаратной спецификации. Эти ресурсы зависят от того, будем ли мы использовать для реализации нашей аппаратуры программируемую логическую интегральную схему (ПЛИС), специализированную интегральную схему (ASIC) или заказную интегральную схему.

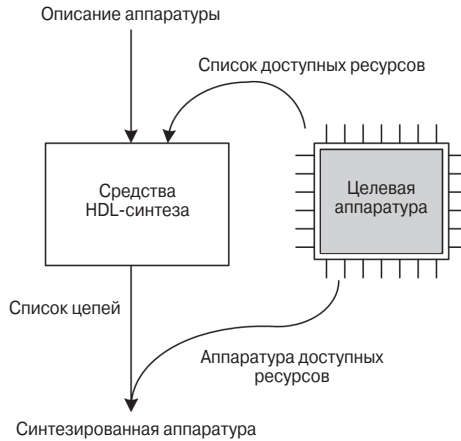


Рис. 1.3. Синтез аппаратной части

Средства синтеза генерируют список соединений библиотечных компонентов из выбранной библиотеки. Список цепей обычно представляется на внутреннем языке списка соединений, языках проектирования VHDL, Verilog или в других форматах списка соединений. Этот список совместно с детальным описанием аппаратной реализации библиотечных элементов формирует законченное пост-синтезное описание (post-synthesis) аппаратной части нашего проекта. Полученное описание может моделироваться как синтезированная аппаратная часть, а также использоваться для программирования ПЛИС или генерации фотошаблонов ASIC.

1.2.7. Компиляция программной части

Для синтеза аппаратной части системы мы должны определить специфику используемой аппаратной платформы. Точно так же компиляция программной части системы должна выполняться для определенного процессора. В процессе компиляции программа на языке высокого уровня превращается в программу в машинных кодах выбранного процессора. Эта машинная программа вместе с аппаратурой процессора, на котором она выполняется, образует законченную реализацию программной части проекта нашей системы. На рис. 1.4 показан процесс компиляции программной части проекта.