Посвящаю моей жене

Введение

Начало нашего века знаменуется бурным развитием и внедрением электронных изделий в повседневную жизнь, а также во все сферы человеческой деятельности (представьте ребенка без электронной игрушки, подростка без смартфона и самолет без бортового компьютера). Однако все эти интересные вещи, прежде чем изготовить, необходимо спроектировать. Поэтому огромная армия инженеров трудится над созданием новых и более совершенных электронных устройств. Канули в лету времена, когда принципиальные схемы сложных устройств чертились вручную или с помощью графических редакторов. На смену им пришли языки проектирования.

Сейчас разработка какого-либо электронного проекта во многом напоминает разработку программ: написание кода программы (написание кода проекта), компиляция программы (компиляция проекта), отладка программы на эталонных примерах (моделирование проекта с помощью специальной программы, называемой *симулятором*), создание исполняемого кода программы (реализация проекта в микросхеме), отладка программы на реальных примерах (физическое моделирование проекта на макетных платах), передача программы заказчику (реализация проекта в виде электронного изделия), сопровождение программы (сопровождение проекта).

В настоящее время наблюдается бурное развитие языков описания аппаратуры (Hardware Description Language — HDL) высокого уровня. Одним из таких языков является язык Verilog. Язык Verilog родился в среде разработчиков цифровых систем, поэтому быстро завоевал популярность у инженеров-практиков. Вскоре язык Verilog стал главным конкурентом языка VHDL. Язык VHDL был разработан по заказу Министерства обороны США, которое всячески способствовало его широкому распространению. На сегодняшний день некоторые программные средства проектирования работают следующим образом: проекты, написанные на языке VHDL, транслируют в язык Verilog, а затем используют компилятор языка Verilog, поскольку язык Verilog более близок к аппаратуре, чем язык VHDL. Отсюда возникает вопрос, не лучше ли сразу создавать проекты на языке Verilog, поскольку любая трансляция требует дополнительных накладных

расходов с точки зрения быстродействия, потребляемой мощности и площади на кристалле, занимаемой проектом.

Однако широкому распространению языка Verilog среди русскоязычных проектировщиков в значительной степени препятствует отсутствие книг по языку Verilog. Те немногочисленные статьи, которые иногда появляются в периодических журналах [1–3] и книга Полякова [4] не могут закрыть указанную проблему. Изучение же языка по стандартам [5–7] весьма затруднительно, поскольку стандарты языка Verilog, прежде всего, предназначены не для пользователей языка, а для разработчиков компиляторов языка. В то же время в мире издано большое количество книг, посвященных языку Verilog, как для начинающих пользователей [8, 9], так и для опытных специалистов [10, 11]. Отметим также, что язык Verilog является объектом серьезных научных исследований [12–21].

В данной книге достаточно полно описываются основные синтаксические элементы и конструкции языка Verilog с точки зрения их практического использования. Каждая конструкция сопровождается примером. Поскольку язык Verilog предназначен как для описания проекта, так и для моделирования проекта, то не все конструкции языка Verilog могут быть реализованы аппаратно, т. е. *синтезированы*. Если конструкция синтезируется, то в книге приводится ее реализация на уровне регистровых передач (Register Transfer Level — RTL). Кроме того, непосредственно в тексте книги предлагается большое количество заданий, выполнение которых способствует более глубокому изучению языка. Некоторые задания посвящены изучению особенностей используемого читателем компилятора. Выполнение таких заданий позволяет приобрести практические знания о возможностях конкретного компилятора. В книге также много замечаний. Они заостряют внимание читателя на отдельных свойствах языка Verilog, его реализации в компиляторах и др. Все замечания следует внимательно прочитать и переосмыслить.

Изложение материала данной книги не привязывается к определенной элементной базе или конкретному программному средству проектирования. Другими словами, материал книги может использоваться при разработке проектов как на заказных СБИС и БМК, так и на ПЛИС. Для демонстрации результатов реализации примеров используется пакет Quartus II версии 12.1 фирмы Altera, однако читатель может использовать любой другой пакет.

Книга имеет следующую структуру. В самом начале, в главе 1, дается быстрое введение в язык Verilog, где на простом примере показан процесс разработки проектов на основе языка Verilog, от описания проекта до его моделирования, а также показаны возможности языка Verilog при разработке сложных иерархических проектов. Затем

приводятся базовые элементы языка. Таким образом, прочтение только первой главы уже позволяет читателям разрабатывать простые проекты.

Главными конструкторскими единицами языка Verilog являются модули, им посвящена глава 2. Чтобы в очередной раз «не изобретать велосипед», можно воспользоваться уже готовыми решениями, такими, как примитивы и библиотечные модули, которые описываются в главе 3. Каждый язык имеет свои типы данных, с которыми он оперирует. Типы данных языка Verilog имеют свои специфические особенности и рассматриваются в главе 4. Использование языка проектирования невозможно без знаний операций, описываемых в главе 5. Особую роль в языке Verilog играет оператор непрерывного назначения, рассматриваемый в главе 6. Функционирование проекта в языке Verilog описывается в виде совокупности взаимодействующих между собой процессов, которые представляют процедурные блоки, рассматриваемые в главе 7. При моделировании важнейшую роль играет время функционирования отдельных частей проекта и всего проекта в целом. Операторы управления процедурным временем представлены в главе 8.

И вот все готово для описания алгоритма функционирования проекта во времени. Для этого служат операторы процедурного назначения, рассматриваемые в главе 9, и операторы процедурного программирования, описываемые в главе 10. Важную роль в языке Verilog также играют атрибуты, представленные в главе 11. С помощью атрибутов пользователь может передать указания (и параметры) компилятору как выполнять компиляцию отдельных фрагментов кода. Блоки генерации, рассматриваемые в главе 12, позволяют сократить описание повторяющихся или незначительно отличающихся фрагментов кода. Задачи и функции языка Verilog, описываемые в главе 13, во многом подобны подпрограммам и функциям языков программирования. Блок спецификаций, рассматриваемый в главе 14, позволяет определять значения временны́х параметров, которые могут использоваться при работе симулятора и временно́го анализатора. Системные задачи и функции приводятся в главе 15, а директивы компилятора — в главе 16. При разработке сложных проектов различными проектировщиками для согласования местоположений отдельных частей проекта служат конфигурации и конфигурационные блоки, рассматриваемые в главе 17. Синтезируемые конструкции языка Verilog приводятся в главе 18.

Настоящая книга, в первую очередь, предназначена разработчикам цифровых устройств и систем для самостоятельного изучения языка Verilog. Поскольку изложение основных элементов и конструкций языка достаточно полное, книга может также использоваться в

качестве справочника языка Verilog. Кроме того, материал книги может быть использован преподавателями для чтения лекций и проведения практических занятий, а также, безусловно, книга предназначена для студентов соответствующих специальностей в качестве учебного пособия при подготовке к практическим занятиям, экзаменам, при написании курсовых и дипломных работ.

Автор выражает искреннюю благодарность сотрудникам фирмы Атомтех (г. Минск, Республика Беларусь), которые прослушали курс лекций на основе рукописи данной книги, за многочисленные вопросы и замечания, способствовавшие улучшению материала книги.