

硬件描述语言 **Verilog**

(第四版)

Donald E. Thomas, Philip R. Moorby 著

刘明业 蒋敬旗 刁岚松 等译

清华大学出版社

KLUWER ACADEMIC PUBLISHERS

(京)新登字 158 号

内 容 简 介

Thomas 和 Moorby 合写的《硬件描述语言 Verilog》已经成为 Verilog 标准的参考书。

这次修改的入门辅导部分通过示例讲述了该语言。这些示例表现了几种重要的描述风格,包括:结构模型、用于逻辑综合的组合电路和时序电路的行为模型、FSM 数据通道模型以及周期精确的描述。行为综合是新添加的一章,讲述了怎样使用这些方式来实现周期精确的描述。

对那些有兴趣描述、模拟和综合数字系统的工程师和学生来说,《硬件描述语言 Verilog》(第四版)是一本很有价值的参考书。

本书是为大学课程编写的。描述风格的介绍顺序符合典型的入门课程的要求(结构的、可综合的、FSM 数据通道的、周期精确的)。本书有一个像学习手册一样的附录。为了有助于体系结构课程的教学,书中还给出了一个简单流水线处理器的模型。

光 盘 介 绍

本书附带一张光盘。其中包含 Veriwell™ Verilog 模拟器、Synplicity Synplify FPGA 综合软件、纯文本的和 PDF 格式的书中示例以及 PDF 格式的讲义。综合工具的使用是有时间限制的,可以根据提示去获取永久性使用权。模拟器和综合工具可在多个平台上使用。

The Verilog Hardware Description Language (Fourth Edition)

Donald E Thomas, Philip R . Moorby

Copyright 1998 by Kluwer Academic Publishers

All Rights Reserved .

本书中文简体字版由 Kluwer Academic Publishers 出版公司授权清华大学出版社独家出版、发行。未经出版者书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

北京市版权局著作权合同登记号:图字 01-2001-1569 号

书 名: 硬件描述语言 Verilog (第四版)

作 者: Donald E . Thomas, Philip R . Moorby 著 刘明业 蒋敬旗 刁岚松 等译

出版者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

[http:// www .tup .tsinghua .edu .cn](http://www.tup.tsinghua.edu.cn)

印刷者: 清华大学印刷厂

发行者: 新华书店总店北京发行所

开 本: 787 × 960 1/ 16 印张: 20 字数: 411 千字

版 次: 2001 年 8 月第 1 版 2001 年 8 月第 1 次印刷

书 号: ISBN 7-900635-36-X

印 数: 0001 ~ 6000

定 价: 36 .00 元

译 者 序

数字集成电路在过去近 30 年的时间里得到迅速发展,EDA 技术在设计过程中起着至关重要的作用。硬件描述语言(HDL)采用形式化的方法,可以直观准确地描述数字电路,应用于模拟验证、设计综合等设计过程中。在众多的硬件描述语言中,Verilog 逐渐发展成为标准的硬件描述语言,是当前 HDL 设计方法学的基础。

硬件描述语言 Verilog 是本书的作者之一——Philip R .Moorby 于 1983 年在英格兰阿克顿市的 Gateway Design Automation 硬件描述语言公司设计出来的,用于从开关级到算法级的多个抽象设计层次的数字设计的建模。该语言提供了一整套功能强大的基元集,包括逻辑门和用户定义的基元;并提供了丰富的结构,这些结构不仅用于硬件的并发性行为的建模,而且用于硬件的时序特性和结构的建模。也可以通过编程语言接口(PLI)对该语言进行扩展。Verilog 语言从诞生起就与生产实际紧密结合在一起,具有结构清晰、文法简明、功能强大、高速模拟和多库支持等优点,并获得许多工具的支持,深受用户的喜爱。据报道,全世界近 90% 的半导体公司都使用硬件描述语言 Verilog。Verilog 实际上是 IC 行业标准,特别是在 1995 年 12 月被 IEEE 接纳为正式标准后,它成为一种很有竞争力的硬件描述语言。

在国内,国家技术监督局于 1998 年正式将硬件描述语言 Verilog 列入国家标准制定项目。本书是 Verilog 的经典著作,由 Verilog 的发明人撰写。本书的翻译工作是配合制订国家标准《集成电路/ 计算机硬件描述语言 Verilog》(国家标准编号为 GB/ T18349-2001,2001 年 10 月 1 日实施)来进行的。在国家标准制订和本书的翻译过程中,得到国内众多专家的帮助和指导,译者在此向北京华大集成电路设计中心王正华研究员,清华大学薛宏熙教授,中科院计算所林宗楷研究员,北京华虹集成电路设计中心李云岗研究员,中科院半导体所薄建国研究员,北京微电子技术研究所王隆望研究员,林守勋研究员,北京理工大学韩月秋教授及陈禾博士表示衷心的感谢。

参加本书翻译的有:蒋敬旗(译前言、第 1、3、4、5、9、10 章),刁岚松(译第 2 章及附录),李杰(合译第 4 章),李春(合译第 5 章),王娟(译第 6 章),吕秀锋(译第 7、8 章)。

全书由刘明业、蒋敬旗和刁岚松统校。

我们在翻译过程中力求翻译准确,但限于译者的水平,一定存在错误和不足之处,恳请读者批评指正。

译 者

2001 年 5 月

前 言

Verilog 语言是一种在广泛的抽象层次设定说明数字系统的硬件描述语言。这种语言支持早期的行为级抽象设计概念,以及后期结构级抽象设计的实现。它包括层次式结构,从而允许设计人员控制描述的复杂度。

Verilog 最初于 1983 年至 1984 年之间的冬季被设计为一种专用的验证/模拟工具。后来,基于这种语言又开发了其他几种专用分析工具,包括故障模拟器和时序分析器。最近 Verilog 也为逻辑综合工具和行为综合工具提供了输入规范。Verilog 语言有助于提供这些工具的一致性。现在,这种语言已标准化为 IEEE # 1364-1995 标准,并且广泛用于多种工具。本书介绍 Verilog 语言,为该语言的初学者和高级用户提供素材。

有时很难将该语言和模拟工具分离开,原因是该语言的动态特性是由模拟器的工作方式确定的。而且,将它和综合工具分离开也是困难的,因为它的语义是由综合工具所允许的输入规范以及所产生的实现来限定的。我们尽可能避开专用模拟器和专用综合器的细节而集中考虑设计的规范。但是,书中包含的内容足以写出可以工作的可执行模型。

本书采用辅导的方法来介绍 Verilog 语言。首先我们从辅导入门开始,通过举例介绍 Verilog 语言的主要特点和进行系统描述的一般方式。接着详细介绍该语言结构,给出大量的示例,使读者通过这些示例可以更容易地学习和复习。最后,在附录中给出 Verilog 语言的形式化描述。总之,我们的方法是通过研讨举例和做练习题来提供一种学习方式。

我们还给出一些练习题供阅读时思考,并建议读者尽可能早地借助 Verilog 模拟器来试试这些练习题。如果你有自己的设计,也可以进行彻底检验。书中所举的例题以电子信号的形式存储在密封的 CD 上。另外,还可参阅 <http://www.ece.cmu.edu/~thomas>。CD 上还包括一个模拟器和一个逻辑综合工具,模拟器所处理的描述规模有限,综合工具给出一个时间不很长的综合演示。

本书的大部分内容假定读者具备初步的逻辑设计和软件编程知识。因此,本书适合于进行集成电路设计的工程师、研究生及电子或计算机工程专业的学生。辅导入门中的内容适于作为逻辑设计的入门课程。它首先介绍结构化组合电路,接着介绍更复杂的可综合的组合电路和时序电路,最后介绍周期准确的系统规范。附录中嵌入辅导入门的内容,给出所讨论的一般错误及辅导入门中要解决的练习题。本书也适合作为高层次逻辑设计、集成电路设计、计算机体系结构及计算机辅助设计(CAD)课程的参考教材。本书为设计课程介绍了覆盖该语言的全部内容,为 CAD 课程介绍了模拟器是怎样工作的。

本书共有十章和七个附录。第 1 章是 Verilog 语言的辅导入门。第 2 章和第 3 章讨论该语言的行为建模方式。第 4 章为逻辑级建模。第 5 章包括定时驱动模拟和事件驱动模拟的高级课题。第 6 章和第 7 章介绍语言在综合中的应用。第 8 章和第 9 章讲述更高级的课题:用户自定义的基元和开关级建模。第 10 章推荐两个可以用作大学教学使用的 Verilog 工程。附录 A 为初学者介绍了辅导入门的内容,附录的其他部分是出现在语言手册中更枯燥的问题,读者可根据自己的需要选择阅读。

祝愿您在设计大型系统时获得乐趣。

您永远的朋友

Donald E . Thomas, Philip R . Moorby

致 谢

作者向维护和促进 Verilog 标准的 Open Verilog International ([http:// www .ovi .org](http://www.ovi.org)) 以及对该语言的不断开发做出贡献的 CAD 工具开发人员和系统设计人员表示感谢。特别是,感谢 Leigh Brady 帮助审阅了初稿,感谢 Elliot Mednick 帮助组织了随书所附 CD 中的内容。

作者也感谢 JoAnn Paul 的帮助及对入门介绍和 CD 内容提出的建议,感谢 John Langworthy 帮助收集附录 A 中的辅导材料,感谢 Tom Martin 帮助整理了第 10 章中的练习题,感谢 H . Fatih Ugurdag 提供了例 7 4。我们也感谢使用本书并提供反馈信息的工程师、教师和学生。最后,感谢 Margaret Hanley 设计了封面和版式。

IEEE 通过了 Verilog 硬件描述语言标准(IEEE # 1364-1995),感谢他们认可了附录 G 中语法的形式化规范。标准的副本可以从 [http:// standard ieee .org](http://standard.ieee.org) 获得。

目 录

第 1 章 Verilog 语言入门辅导	1
1.1 开始	1
1.1.1 结构描述	1
1.1.2 模拟 binaryToESeg 驱动源	3
1.1.3 为模块建立端口	5
1.1.4 为模块建立测试台	6
1.2 组合电路的行为建模	9
1.2.1 过程模型	9
1.2.2 综合组合电路的规则	11
1.3 时钟时序电路的行为建模	11
1.3.1 建立有限状态机模型	12
1.3.2 综合时序系统的规则	15
1.3.3 非阻塞赋值(“ <= ”)	15
1.4 模块的层次	17
1.4.1 计数器	17
1.4.2 系统时钟	18
1.4.3 将整个电路结合在一起	19
1.4.4 将行为模块和结构模块连接在一起	22
1.5 有限状态机和数据通道	23
1.5.1 简单计算示例	24
1.5.2 系统的数据通道	24
1.5.3 数据通道功能模块的细节	25
1.5.4 用连线将数据通道连在一起	27
1.5.5 FSM 说明	28
1.6 周期精确的行为描述	32
1.6.1 规范方法	33
1.6.2 几点注释	35
1.7 赋值语句的总结	35
1.8 小结	37

1.9	练习.....	37
第2章	行为建模	39
2.1	进程模型.....	39
2.2	If-Then-Else	40
2.2.1	else 如何与 if 语句配对	44
2.2.2	条件操作符	45
2.3	循环语句.....	46
2.3.1	四种基本循环语句	46
2.3.2	循环的异常退出	49
2.4	多分支语句.....	50
2.4.1	If-Else-If	50
2.4.2	Case	50
2.4.3	Case 和 If-Else-If 的比较	53
2.4.4	Casez 和 Casex	53
2.5	函数和任务.....	54
2.5.1	任务	56
2.5.2	函数	59
2.5.3	结构视域	61
2.6	作用域规则和层次名.....	63
2.6.1	作用域规则	64
2.6.2	层次名	66
2.7	小结.....	66
2.8	练习.....	67
第3章	并发进程	69
3.1	并发进程.....	69
3.2	事件.....	70
3.2.1	事件控制语句	71
3.2.2	有名事件	72
3.3	等待语句.....	75
3.3.1	一个完整的生产者和消费者握手示例	76
3.3.2	Wait 语句和 While 语句的对比.....	79
3.3.3	Wait 语句和事件控制语句的比较	80

3.4	并发进程示例.....	80
3.5	简单流水线处理器.....	86
3.5.1	基本处理器	86
3.5.2	流水线之间的同步	88
3.6	有名块的终止.....	89
3.7	赋值语句内部控制和定时事件.....	91
3.8	过程持续赋值.....	94
3.9	顺序模块和并行模块.....	95
3.10	练习	98
第4章	逻辑级建模.....	100
4.1	引言	100
4.2	逻辑门与线网	101
4.2.1	用基元逻辑门建模.....	101
4.2.2	四级逻辑值.....	104
4.2.3	线网.....	105
4.2.4	模块例示与端口规范.....	108
4.2.5	有关逻辑级的一个示例.....	109
4.3	示例数组	115
4.4	持续赋值	118
4.4.1	组合电路的行为建模.....	119
4.4.2	线网与持续赋值语句声明.....	120
4.5	参数化定义	123
4.6	行为级/结构级的混合示例.....	127
4.7	逻辑延迟建模	131
4.7.1	门级建模示例.....	131
4.7.2	门和线网延迟.....	133
4.7.3	时间单位的规定.....	135
4.7.4	最小延迟、典型延迟和最大延迟	136
4.8	模块中的延迟路径	137
4.9	小结	139
4.10	练习.....	139

第 5 章	高级时序	142
5.1	Verilog 时序模型.....	142
5.2	模拟器的基本模型.....	145
5.2.1	门级模拟.....	145
5.2.2	更通用的模型.....	146
5.2.3	行为级模型的调度.....	148
5.3	模拟算法的不确定行为.....	150
5.3.1	临近不确定区.....	150
5.3.2	Verilog 是一种并发语言.....	152
5.4	非阻塞过程赋值语句.....	155
5.4.1	阻塞和非阻塞赋值的比较.....	155
5.4.2	非阻塞赋值的一般用法.....	156
5.4.3	事件驱动调度算法的扩展.....	157
5.4.4	非阻塞赋值的举例分析.....	159
5.5	小结.....	162
5.6	练习题.....	162
第 6 章	逻辑综合	167
6.1	综合概述.....	167
6.1.1	寄存器传输级系统.....	167
6.1.2	限制声明.....	168
6.2	使用门和持续赋值的组合逻辑.....	168
6.3	用来说明组合逻辑的过程语句.....	170
6.3.1	基础.....	171
6.3.2	复杂形式——推断出的锁存器.....	172
6.3.3	说明无关项.....	174
6.3.4	过程循环结构.....	175
6.4	时序元件的推断.....	176
6.4.1	锁存器的推断.....	176
6.4.2	触发器的推断.....	178
6.4.3	小结.....	180
6.5	三态器件的推断.....	180
6.6	有限状态机的描述.....	181
6.6.1	有限状态机的示例.....	181

6.6.2	FSM 说明的另一种方式	184
6.7	逻辑综合的总结	185
6.8	习题	186
第 7 章	行为综合	188
7.1	行为综合的介绍	188
7.2	周期精确的说明	189
7.2.1	always 块的输入和输出	189
7.2.2	always 块的输入和输出关系	190
7.2.3	复位功能说明	193
7.3	米利 摩尔机的说明	194
7.3.1	复杂控制的说明	195
7.3.2	数据与控制路径的折中	196
7.4	小结	199
第 8 章	用户定义的基元	200
8.1	组合基元	200
8.1.1	用户定义基元的基本特征	200
8.1.2	组合逻辑电路的描述	202
8.2	时序基元	203
8.2.1	电平敏感的基元	204
8.2.2	边沿敏感的基元	205
8.3	速记表示法	207
8.4	电平敏感与边沿敏感混合的基元	208
8.5	小结	210
8.6	练习	210
第 9 章	开关级建模	212
9.1	动态 MOS 移位寄存器示例	212
9.2	开关级建模	216
9.2.1	强度建模	216
9.2.2	强度的定义	218
9.2.3	使用强度的示例	220
9.2.4	电阻型 MOS 门	221

9.3	二义性强度	223
9.3.1	二义性强度的说明	223
9.3.2	基本计算	224
9.4	miniSim 示例	228
9.4.1	概述	228
9.4.2	miniSim 的源码	229
9.4.3	模拟结果	239
9.5	小结	241
9.6	练习	241
第 10 章	工程项目	243
10.1	建立功耗模型	243
10.1.1	对功耗进行建模	243
10.1.2	需要做什么	243
10.1.3	步骤	244
10.2	软盘控制器	245
10.2.1	介绍	245
10.2.2	磁盘格式	245
10.2.3	功能描述	247
10.2.4	真实设备	248
10.2.5	你一直想知道的有关 CRC 的各种情况	249
10.2.6	起支持作用的 Verilog 模块	249
附录 A	学习指南	251
A.1	结构描述	251
A.2	测试台模块	259
A.3	使用 always 的组合电路	259
A.4	时序电路	262
A.5	层次化描述	264
A.6	有限状态机和数据通道	264
A.7	周期精确描述	265
附录 B	词法	266
B.1	空白符和注释	266

B.2	操作符	266
B.3	数字	266
B.4	字符串	267
B.5	标识符、系统名和关键字	268
附录 C	Verilog 操作符	270
C.1	操作符表	270
C.2	操作符优先级	272
C.3	操作符真值表	273
C.4	表达式的位数	273
附录 D	Verilog 门类型	275
D.1	逻辑门	275
D.2	BUF 和 NOT 门	276
D.3	BUFIF 和 NOTIF 门	276
D.4	MOS 门	277
D.5	双向门	278
D.6	CMOS 门	278
D.7	Pullup 和 Pulldown 门	278
附录 E	寄存器、存储器、整数和时间	279
E.1	寄存器	279
E.2	存储器	280
E.3	整数和时间	280
附录 F	系统任务和函数	282
F.1	Display 和 Write 任务	282
F.2	持续监视	283
F.3	选通监视	283
F.4	文件输出	283
F.5	模拟时间	284
F.6	停止和完成	284
F.7	随机函数	285
F.8	从磁盘文件读数据	285

附录 G	形式化语法定义	286
G 1	形式化语法规范指南	286
G 2	源文本	290
G 3	声明	291
G 4	基元示例	293
G 5	模块例示	294
G 6	UDP 的声明和例示	295
G 7	行为语句	296
G 8	Specify 部分	298
G 9	表达式	301
G 10	通用说明	304

第 1 章 Verilog 语言入门辅导

数字系统是非常复杂的。从最基本的层次来看,如果我们把一个系统看作逻辑门或传输晶体管的集合,它们可能由数以百万计的元件组成。从更抽象的层次来看,这些元件可以组成一些功能部件,如高速缓存、浮点部件、信号处理器或实时控制器等。硬件描述语言已经发展起来,用来辅助设计具有大量元件、从电路级到逻辑抽象级诸多层次的系统。

数字系统的设计过程是先建立逻辑系统设计的概念、最终实现必须满足的一组约束以及建立系统的一组基本元件。设计是一个先用手工做或者先用自动综合、然后再根据给出的约束进行测试的迭代过程。一个设计一般可划分为许多更小的部分(根据众所周知的分治工程方法),而各部分可以再划分,直到整个设计用已知的基本元件说明为止。

Verilog 语言为数字系统设计人员提供了一种在广泛的抽象层次上描述数字系统的方式,同时,在这些层次上为计算机辅助设计工具在工程设计中进行辅助设计提供了方法。该语言支持早期的行为结构设计概念,以及其后层次化结构设计的实现。在设计过程中,进行逻辑结构设计部分时可以将行为结构和层次化结构混合起来。为确认正确性可以将描述进行模拟,也有一些用于自动设计的综合工具。Verilog 语言为设计者进行大型复杂的数字系统设计提供了途径。本章概括介绍了 Verilog 语言的基本特点。

1.1 开始

Verilog 语言将一个数字系统描述为一组模块。每个模块与其他模块及其本身的描述内容都有一个接口。一个模块代表一个逻辑单元,可以通过规定其内部逻辑结构来进行描述——例如描述实际的逻辑门,或者通过用像程序一样的方式来描述它的行为——在这种情况下主要考虑模块所完成的功能而不是其逻辑实现。然后将这些模块互连起来,使它们能够互相通信。

1.1.1 结构描述

首先介绍初级逻辑设计过程中的一个基本逻辑电路:一个二进制七段显示驱动源,如例 1.1 所示。显示驱动源具有一个 4 位二进制输入,驱动七个段显示数字 0 至 9 及十六进制数 A 到 F。本例中所示的只是驱动 E 段的逻辑。

例 1.1 二进制七段显示驱动源(仅仅 E 段)

```

module binaryToESeg;
    wire      eSeg, p1, p2, p3, p4;
    reg       A, B, C, D;

    nand # 1
        g1 (p1, C, ~ D),
        g2 (p2, A, B),
        g3 (p3, ~ B, ~ D),
        g4 (p4, A, C),
        g5 (eSeg, p1, p2, p3, p4);
endmodule

```

例 1.1 中也给出了这个电路的 Verilog 描述。描述表示的是一个模块的基本定义，模块名为 **binaryToESeg**。每个模块定义包括关键字 `module`，紧随其后的模块名，以及最后的 `endmodule` 语句。描述中的第二行规定在这个模块的子模块间传送逻辑值的 `wire` 的名称。第三行说明保持逻辑值的存储单元名称。这些寄存器是触发器电路元件的一个抽象。

第 5 行及随后的第 6 行到第 10 行示例化(instantiate)了 5 个与非门，每个与非门有一个单位时间的延迟。与非门是语言中的一个预定义逻辑门类型，其他类型包括 AND、OR 及 XOR，后边会有详细介绍。这个语句规定电路中的 5 个门，称作 `g1` 到 `g5`。“# 1”表示每个门都有一个单位时间的延迟。最后，括号内的符号表示连接这些门的连线和寄存器。括号内的第一个符号是门的输出，其他符号是输入。NOT 的运算符(“~”)用来规定连接到输入端逻辑值的补码。为进一步说明逻辑图和与之等效的 Verilog 描述之间的一致性，原理图中包括连线、寄存器及示例名称。

尽管这个示例简单，但它列举出 Verilog 语言的几个要点。首先是模块定义与模块例示的概念。使用模块语句，如上例所示，一旦规定了所有的内部细节就定义了一个模块。而后这个模块可以在设计中多次使用(例示)。每个例示称为模块的一个示例，可以分别命名和连接。基元门，如与非门，是语言提供的预定义逻辑基元。第 4 章内有更详细的介绍。

各种逻辑门是由线网(net)连接起来的。线网是语言中两个基本数据类型中的一个(寄存器是另外一种数据类型)，用来模拟像门这样的结构化实体间的连接。连线(wire)是 net 的一种类型，其他类型包括 wired-AND 连接、wired-OR 连接及 trireg 连接。不同的 net 类型分别第 4 章和第 9 章中有更详细的描述。

我们知道多个逻辑门以层次化方式可以构成更大的模块。在本例中，用与非门来构造 `binaryToESeg` 模块。如果这个 `binaryToESeg` 模块有输入输出端口，通过将它例示成另一个模块就可以作为更大模块中的一部分，依此类推。通过将设计划分为更小更有意

义的部分(即子模块),利用层次化描述可以控制设计的复杂度。在例示子模块时,必须知道各子模块的接口,而子模块复杂的实现细节在其他地方描述,因此不用出现在当前模块的描述中。

最后,应当指出 A、B、C、D 命名为寄存器显得不规则。读者可能会认为它们应当是模块 `binaryToESeg` 的输入端,而 `eSeg` 值应当是一个输出端,最终它们会变成输入端和输出端。但是现在我们仍保持这些寄存器定义,并且它们会在下一节的模拟中发挥作用。

参考: 门基元 4.2.1, net 规范 4.2.3。

1.1.2 模拟 `binaryToESeg` 驱动源

例 1.2 给出了一个更完整的 `binaryToESeg` 模块定义,称作 `binaryToESegSim`。这个示例包括为 NAND 门示例提供激励的语句,以及监视其输出端变化的语句。尽管没有给出所有可能的输入组合,但给出的这些输入组合将解释怎样提供输入激励。

例 1.2 用于模拟的 `binaryToESeg` 驱动源

```
module binaryToESegSim;
    wire      eSeg, p1, p2, p3, p4;
    reg       A, B, C, D;

    nand #1
        g1 (p1, C, ~D),
        g2 (p2, A, B),
        g3 (p3, ~B, ~D),
        g4 (p4, A, C),
        g5 (eSeg, p1, p2, p3, p4);

    initial // two slashes introduce a single line comment
        begin
            $monitor ($time, , ,
                A = %b B = %b C = %b D = %b, eSeg = %b ,
                A, B, C, D, eSeg);
            // waveform for simulating the binaryToESeg driver
            #10 A = 0; B = 0; C = 0; D = 0;
            #10 D = 1;
            #10 C = 1; D = 0;
            #10 $finish;
        end
endmodule
```