第3章 iverilog+GTKWave 下载、安装及使用

3.1 iverilog 和 GTKWave 简介

iverilog 的全称为 icarus Verilog,它是一款轻量级开源的 IEEE-1364 Verilog 硬件描述语言的免费编译软件,用于数字和模拟电路的仿真与验证。iverilog 是基于 C++ 实现的,开发者是 Stephen Williams,遵循 GNU GPL license 许可证。

目前,由于 EDA 三巨头 Synopsys、Cadence、Mentor 所提供的 Verilog 编译与仿真软件的版权问题,国内外很多高校和企业在数字芯片的教学与设计中都采用 iverilog,其被 戏称为"全球第四大"数字芯片仿真器。虽然,iverilog 的生态支持度不如三大主流的仿真 工具,但是其开源性和便捷性被越来越多的用户青睐。

iverilog 的主要功能特点如下。

(1) 编译速度快,并支持包括 Verilog 1995、2001、2005、SystemVerilog 2009 在内的 多种 Verilog 语言。

(2) 支持多种平台,轻量级且安装方便,iverilog 支持包括 Windows、Linux、macOS 在内的多种操作系统,且轻量级(iverilog 包含 GTKWave 的安装文件大约 10MB)安装速度快。

(3) 支持 VCD 文件格式: iverilog 可以生成 vcd 波形文件,方便用户查看仿真结果,GTKWave 就是用来查看 vcd 波形文件的。

iverilog 的主要缺点如下。

(1) 没有 Verilog 语言编辑界面,需要 VS Code 等集成开发环境来编写代码,编写过程中的纠错能力不强。

(2)编译器的编译错误反馈信息不是很详细,定位也不是很准确,导致不易发现及改正程序中的错误。

GTKWave 是一种分析工具,它主要用于对 Verilog 或 VHDL 仿真模型进行调试。 GTKWave 用于在大型芯片上进行调试任务,并作为第三方的离线纠错工具。它是一个 64 位的独立软件,可以运行在具有足够物理内存的工作站上。除了可以交互式地观看 VCD 文件外,GTKWave 并不支持与模拟仿真软件的交互式运行,它通过使用转储文件 来查看模拟与仿真后的波形分析,支持多种转储文件格式,主要包括 VCD、LXT、LXT2、 VZT 等文件格式。其中,VCD 是指 Value Change Dump 值变化转储,这是由大多数 Verilog 模拟器生成的行业标准文件格式,并由 IEEE-1364 指定,是查看器处理速度最 慢、需要占用最多内存的格式。但由于这种格式的普遍性,而且几乎所有模拟器都支持 它,所以 GTKWave 保留了 VCD 文件支持。

3.2 iverilog+GTKWave 下载及安装

iverilog 的下载网站为 http://bleyer.org/icarus/,该网站提供面向 Windows 系统的 多个 iverilog 下载版本,推荐下载图 3.1 中标注的稳定版。

Icarus Verilog for Windows	
Icarus Verilog is a free compiler implementation for the IEEE-1	364 Verilog hardware description language. Icarus is mainta
In this page you will find easy to install Icarus Verilog package Inno Setup free installer utility.	es compiled with the MinGW toolchain for the Windows envir
Download	
You can find Icarus Verilog sources and binaries for most platfe	orms at the Icarus site FTP. The sources available here have
 iverilog-v12-20220611-x64_setup [18.2MB] iverilog-v11-20210204-x64_setup.exe [44.1MB] iverilog-10.1-20201123-x64_setup.exe [18.1MB] iverilog-10.0-x86_setup.exe [9.77MB] iverilog-20.0-x86_setup.exe (development snapshot) [iverilog-0.9.7_setup.exe (latest stable release) [10.5ME iverilog-0.9.7_setup.exe [10.4MB] 	11.2MB] Ц
 iverilog-0.8.6_setup.exe (latest release 0.8 series) [1.2 iverilog-0.7-20040706_setup.exe [1.09MB] iverilog-0.7 	9MB] <u>iverilog-0.8.6.7z [800kB]</u> 20040706.7z [588kB]
图 3.1 iverilo	g下载页面

比较好的情况是,iverilog 的安装文件中自带 GTKWave,所以只需要下载 iverilog 并 安装即可,与此同时也安装 GTKWave。在下载 iverilog 安装程序后,直接双击程序开始 安装,iverilog 在安装过程中只需要设置安装路径就可以轻松完成,如图 3.2 所示,iverilog 安装过程中需要同时安装 GTKWave。

🕼 Setup - Icarus Verilog	-		X
Select Components Which components should be installed?		[R
Select the components you want to install; clear the components you install. Click Next when you are ready to continue.	do not	want to	
Full installation			~
☑ Install GTKWave		34.5 M	IB
Current selection requires at least 47.8 MB of disk space.			
< <u>B</u> ack <u>N</u> ext	>	Ca	ncel

图 3.2 iverilog 安装过程

在完成程序的安装后,会得到下列文件,如图 3.3 所示。

名称	修改日期	类型	大小
📕 bin	2023/7/26 10:28	文件夹	
📕 include	2023/7/26 10:28	文件夹	
📕 install	2023/7/26 10:28	文件夹	
📕 lib	2023/7/26 10:28	文件夹	
samples	2023/7/26 10:28	文件夹	
share	2023/7/26 10:28	文件夹	
🐢 Icarus Verilog	2023/7/26 10:28	Internet 快捷方式	1 KB
🕘 icarus	2004/4/25 16:52	图标	19 KB
📾 iverilog	2013/8/20 16:41	Microsoft Edge PD	23 KB
🧱 iverilog-vpi	2013/8/20 16:41	Microsoft Edge PD	8 KB
📄 unins000.dat	2023/7/26 10:28	DAT 文件	145 KB
🏶 unins000	2023/7/25 22:18	应用程序	1,180 KB
👼 vvp	2013/8/20 16:41	Microsoft Edge PD	10 KB

图 3.3 iverilog 安装完成

安装完成后,在iverilog使用前还需要设置环境变量,即在系统环境变量 path 中加入 图 3.3 中的 bin 文件的路径,例如 C:\iverilog\bin,如图 3.4 所示。

环境变量		×
Saga	编辑环境变量	×
变i Or	C:\Program Files (x86)\Common Files\Oracle\Java\javapath	新建(<u>N</u>)
Pa TEI	C:\Program Files (x86)\Intel\(CLS Client\ C:\Program Files\Intel\iCLS Client\ %SystemRod%System32	编辑(E)
IM	%SystemRoot% %SystemRoot%\System32\Wbem	浏览(B)
	%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\ %SYSTEMROOT%\System32\OpenSSH\	删除(D)
	C:\iverilog\bin	上移(U)
系统		下移(Q)
变i Co		^
Dri LM		编辑文本([]
OS Pai		
PA		~
	确定	取消
	The State St	
	明正	W/H

图 3.4 环境变量设置

在设置环境变量后,可以在命令提示符中输入 iverilog,如果安装成功,会得到安装信息,如图 3.5 所示。

为验证是否成功安装了 GTKWave,可以在命令提示符中输入 gtkwave,如果成功安装,则 GTKWave 的窗口会直接弹出。

GTKWave的窗口包括菜单栏、工具栏、信号源选择区、波形显示区,如图 3.6 所示。

(1) 菜单栏提供的功能包括: File(文件)、Edit(编辑)、Search(查找)、Time(时间



信号源选择区

图 3.6 GTKWave 界面

波形显示区

- 轴)、Markers(标记)、View(视图)、Help(帮助)。
 - File(文件)包括: Open(打开)、Save(保存)、Export(导出)等功能。
 - Edit(编辑)包括: Insert(插入)、Data Format(数据格式)、Color Format(颜色格 式)等功能。
 - Search(查找)包括: Pattern Search(模式查找)、Signal Search(信号查找)等查找功能。
 - Markers(标记)包括: Collect Markers(收集标记)、Drop Markers(放弃标记)等与标记相关操作的功能。
 - View(视图)包括: Show Mouseover(显示鼠标轨迹)、Show Wave Highlight(高 亮显示波形)等方便查看波形的功能。
 - (2) 工具栏提供包括放大、缩小等一系列操作工具。
 - (3) 信号选择区在打开 vcd 文件后会显示相应的变量名称。

(4) 波形显示区用来查看不同变量的波形。

3.3 iverilog+GTKWave 使用实例

本节通过实例进一步熟悉 iverilog+GTKWave 的使用方法。该实例引自网络^[1]。

下面是一个周期 m = 15 的编码结构和移存器状态。它使用 $f_1(x) = x^4 + x + 1$ 来构造 m 序列产生器,其编码器结构和移存器状态如图 3.7 所示。图中每一列都对应一个 m 序列,第一列对应的 m 序列为

 $A_1 = (a_0 a_1 \cdots a_{m-1}) = (111101011001000)$

用码多项式形式表示为

 $A_{1}(x) = a_{0} + a_{1}x + \dots + a_{m-1}x^{m-1} = 1 + x + x^{2} + x^{3} + x^{5} + x^{7} + x^{8} + x^{11}$



根据上述描述,分别编写两个程序来模拟实现功能的源文件 prbs4.v 和测试使用的 testbench 文件 prbs4_tb.v。

```
prbs4.v
module prbs4(clk, rst_n, out, out_invalid_n);
    input clk;
    input rst_n;
    output reg out;
```

[1] m0_46521579的博客."Icarus/erilog." 2023-01-18.

 $\label{eq:https://blog.csdn.net/m0_46521579/article/detais/128728621? ops_request_misc = \% 257B\% 2522request\% 255Fid\% 2522\% 253A\% 2522189211d671a6e3e7bbcba33dc7ba7ab8\% 2522\% 252C\% 2522scm\% 2522\% 253A\% 25220140713.130102334.pc\% 255Fblog.\% 2522\% 257D\&-request_d = 189211d671a6e3e7bbcba33dc7ba7ab8\&-biz_id=0\&.utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-1-128728621-null-null.nonecase\&.utm_term=iverilog&.spm = 1018.2226.3001.4450.$

```
output reg out invalid n;
    reg[3:0] a;
    always@(posedge clk or negedge rst_n) begin
        if(!rst n) begin
           a <=4'b1000;
        end
        else begin
           a[3] <=a[3]^a[0];
           a[2] <= a[3];
           a[1] <=a[2];
           a[0] <= a[1];
        end
    end
    always@(posedge clk or negedge rst n) begin
        if(!rst n) begin
           out <=1'b0;
           out invalid n <=1'b0;</pre>
        end
        else begin
           out <=a[0];
           out_invalid_n <=1'b1;</pre>
        end
    end
endmodule
prbs4 tb.v
`timescale 1ns / 1ps
module prbs4 tb();
reg clk;
reg rst n;
wire out;
wire out valid n;
integer i;
prbs4 uut(clk, rst n, out, out valid n);
initial begin
    $dumpfile("prbs4.vcd");
    $dumpvars(0, prbs4 tb);
end
initial begin
   rst n =1'b0;
    #20 rst_n =1'b1;
end
initial begin
   clk = 0;
    for(i=0; i<45; i=i+1) begin
       #1 clk =1;
        #1 clk =0;
    end
    $display("test complete");
end
endmodule
```

我们使用 vscode 输入和保存上述代码,如图 3.8 所示的实例程序代码。



图 3.8 实例程序代码

在编写程序时要注意下边两句,这是 testbench 文件中 iverilog 编译器专用于生成波 形文件的语句。\$ dumpfile 指定输出的波形文件名称, \$ dumpvars 指定将哪些变量的信 息保存到 \$ dumpfile 指定的 VCD 文件中。

在完成输入后,进入命令提示符,首先进入程序保存的文件夹,本书是将文件存储在 C:\test\目录下的,因此,在进入相应文件夹下后输入

iverilog -o prbs4 prbs4.v prbs4 tb.v

其中,第一个"-o prbs4"代表输出文件名为 prbs4.out,输出默认为 a.out 文件,在编译完成后,如果没有任何反应就是编译通过了,然后在文件夹中会有一个名为 prbs4 的文件。为了产生波形文件,我们继续在命令行中输入:

vvp -n prbs4 -1xt2

vvp 命令是用来输出波形文件的,如果输入后提示 LXT2 info: dumpfile prbs4.vcd opened for output. test complete,就表明波形文件生成成功了,生成的波形文件名为 prbs4.vcd。最后就可以通过 gtkwave 来查看波形文件了。在命令提示符中输入: gtkwave prbs4.vcd,运行后就会打开如图 3.9 所示的界面。

其中,在SST 区域有 testbench 文件和在 testbench 文件中实例化的各个实体,在本

实体	已选变量			
GTKWave - prbs4.vcd				- 🗆 X
File Edit Search Time	Markers View Help			
💥 🕞 🙍 í 🔍 🔍 🤄	2. 🛬 🕼 🗐 🖨 🖨 Fro	om: 0 sec To: 90 ns	🔁 Marker: Cursor: 1 pr	5
* <u>S</u> ST	Signals Waves	1.84	2 44	1
Signals clt[0] (31:0) out_valid_n[0] rst_n[0]				
Filter:	9 e 7 e			<u>+</u>
~				



例中,我们只有一个 prbs4.v 文件的实体叫作 uut,单击不同的实体,会在 Signals 框中显示该实体或 testbench 中声明的变量。用户如果需要看某个变量的波形图,则需要将该变量从变量区拖曳到其右侧的已选变量区。我们选择 uut 实体的 out[0]变量和 clk 变量来查看它们的波形,如图 3.10 所示。

GTKWave - prbs4.vcd		-		\times
File Edit Search Time Markers View	Help			
x 🗗 🖸 I 🔍 Q Q 🛬 🎼 🖨	🛛 🗍 🖕 🛶 🗍 From: 0 sec 👘 To: 90 ns 👘 🖓 Marker: Cursor: 36190 ps			
Signals	Waves			
Time Clk[0] Cut[0] Cut[0]			70 ns	^
Signals				
clk[0]				
i[31:0]				
out[0]				
out_valid_n[0]				
Filter:				4
Append Insert Replace				•

图 3.10 prbs4 的波形文件

通过合适的放大和缩小,我们可以看到 clk 持续地发射脉冲信号, 而 out 则可以按照 之前的要求进行信号的输出。

3.4 实 验

本节使用 iverilog+GTKWave 完成运算器设计。

1. 实验目的

熟悉 iverilog+GTKWave 平台运行环境。

2. 实验要求

(1)运算器可以完成常规的算术以及逻辑运算,包括 addu、subu、ori、addi、andi 等常用的算术逻辑运算指令。

(2) 使用 iverilog 平台实现运算器。

(3) 使用 iverilog 平台完成一个独立的 testbench 设计,并能完成各个指令的测试 功能。

(4) 使用 GTKWave 记录测试过程的各种波形变换,并分析指令完成过程。