





1. 波形文件概述

2. 仿真验证方法:开源工具



波形文件一般用于仿真后记录波形文件,用于做详细分析和研究。

常见的波形文件格式主要有:

- VCD(Value Change Dump),标准波形文件,所有仿真器都必须支持
- fsdb(Fast Signal DataBase), Verdi支持
- WLF(Wave Log File), modelsim产生

各家不同的仿真或调试工具支持的波形文件类型,互不通用,但基本都可以由VCD文件转换而来(其实就是VCD文件的压缩版,因为只取仿真调试需要的数据,所以文件大小要远小于原始VCD文件),有的还提供与VCD文件的互转换功能。

推荐调试方法: 使用各种仿真器后台完成仿真, 生成fsdb波形; 然后使用verdi进行查看波形与调试

VCD是 Verilog HDL语言标准的一部分,因此所有的verilog的仿真器都能够查看该文件,允许用户在verilog代码中通过系统函数来dump VCD文件。

它包含了信号的变化信息,记录了整个仿真的信息。 优点:可以通过VCD文件来估计设计的功耗,而这一点也是其他波形文件所不具备的。 缺点非常明显:体积巨大。 可以通过Verilog HDL的系统函数dumpfile来生成波形,通过dumpvars的参数来规定我们抽取仿真中某些特定模块和信号的数据。

生成方法: 在testbench中加入如下几行代码即可

```
initial
begin
$fsdbDumpfile("*.fsdb"); //*代表生成的fsdb的文件名
$fsdbDumpvars(0,**); //**代表测试文件名
end
```

fsdb文件是verdi使用一种专用的数据格式,类似于VCD,但是它是只提出了仿真过程中信号的有用信息,除去了 VCD中信息冗余。因此,fsdb数据量小,而且会提高仿真速度。 各家的仿真工具如vcs、ncsim, modlesim 等可以通过加载Verdi 的PLI (一般位于安装目录下的share/pli 目录下) 而直接dump fsdb文件。

生成方法:在testbench中加入如下几行代码即可

initial
begin
\$fsdbDumpfile("*.fsdb"); //*代表生成的fsdb的文件名
\$fsdbDumpvars(0,**); //**代表测试文件名
end



1. 波形文件概述

2. 仿真验证方法:开源工具+商业工具

3. 逻辑综合简单流程:开源工具+商业工具

iverilog:

Icarus Verilog是一个verilog仿真工具,以编译器的形式工作,将以verilog编写的源代码编译为某种目标格式.如果要进行仿真的话,它可以生成一个叫做vvp的中间格式.这个格式可以由其所附带的vvp命令执行.

gtkwave:

wave viewer. 可以用于查看标准的verilog VCD/EVCD, 以及其他的一些格式的波形文件.

二者都是开源工具,源代码公开,都可以很容易下载到 建议在Linux系统下安装使用

使用参考 https://zhuanlan.zhihu.com/p/148795858

RTL代码

```
module adder(clk, rst_n, a, b, c);
          input [3:0] a;
          input [3:0] b;
          output [7:0] c;
          input clk, rst_n;
          wire [3:0] a;
          wire [3:0] b;
          reg [7:0] c;
          always @(posedge clk or negedge rst_n) begin
                    if (rst_n = 1'b0)
                              c <= 8'b0;
                    else
                              c <= a+b;
          end
endmodule
```

`timescale 1ns/1ns	initial beg	egin
<pre>module adder_tb();</pre>		$rst_n = 1;$
reg [3:0] a;		test(4'b1111, 4'b1111, 5'b11110);
reg [3:0] b;		\$finish;
wire [7:0] c;	end	
	task test;	• ?
reg clk,rst_n;		input [3:0] in;
<u> </u>		input [3:0] in2;
adder DUT (input [7:0] e;
.clk(clk),		begin
.rst_n(rst_n),		a = in;
.a(a),		b = in2;
.b(b),		@(posedge clk);
.c(c)		@(negedge clk);
);		if $(c == e)$ begin
		\$display("It works");
always begin		end else begin
#10 clk = 0;		\$display("opps %d + %d ~= %d, expect %d", in, in2, c, e);
#10 clk = 1;		end
end		end
	endtask	
	endmodule	

Testbench代码

iverilog命令行编译运行:

> iverilog adder_rtl.v adder_tb.v

```
    → example ls
    adder_rtl.v adder_tb.v a.out*
    → example iverilog adder_rtl.v adder_tb.v
    → example ls
    adder_rtl.v adder_tb.v a.out*
    → example ./a.out
    It works
    → example [
```

在tb中添加dump:

initial begin \$dumpfile("wave.vcd"); // 指定用作dumpfile的文件 \$dumpvars; // dump all vars end

重新编译运行一遍,生成了一个vcd文件,使用gtkwave查看.

```
→ example ls
adder_rtl.v adder_tb.v
→ example iverilog adder_rtl.v adder_tb.v
→ example ls
adder_rtl.v adder_tb.v a.out*
→ example ./a.out
VCD info: dumpfile wave.vcd opened for output.
It works
→ example ls
adder_rtl.v adder_tb.v a.out* wave.vcd
→ example gtkwave wave.vcd
```

File Edit Search Time Markers View Help



File Edit Search Time Markers View Help

🔏 🗄 🛅 🛛 🗗 🖯 🍕	ラ 🕼 🔊 🛛	🚱 🛷 🛛 From: 🛛 sec	To: 50 ns	🛛 🔂 🛛 Marker: Cursor: 14 ns						
<u>▼ S</u> ST	Signals	Waves								
由書adder tb	Time		L5 ns	20 ns	25 ns	30 ns	35 ns	40 ns	45 ns	50
- 👬 DUT	a[3:0]	x				F				
L 🏶 test	c[7:0]	xx						16		
	clk									
	rst_n									
Tuno Cignals	e[7:0]	x				F				
	in2[3:0]	x				F				
reg e[7:0]	:									
reg in2[3:0]										
reg in[3:0]										
Filter										
Append Insert Replace		I								





轻松成为设计高手:VerilogHDL实用精解. EDA先锋工作室. 北京航空航天大学出版社. 2012年.



中国大学MOOC平台: 芯动力——硬件加速设计方法 https://www.icourse163.org/course/SWJTU-1207492806

