Verdi 自己是不能编译也不能仿真的，需要用 VCS 来配合进行编译和仿真。VCS 专注于编译和仿真，Verdi 只是一个波形查看工具。与 VCS 类似的工具还有 Modelsim，Modelsim 集编译、仿真、波形调试于一体，用起来也很方便。

1、准备

1.1 Makefile 脚本

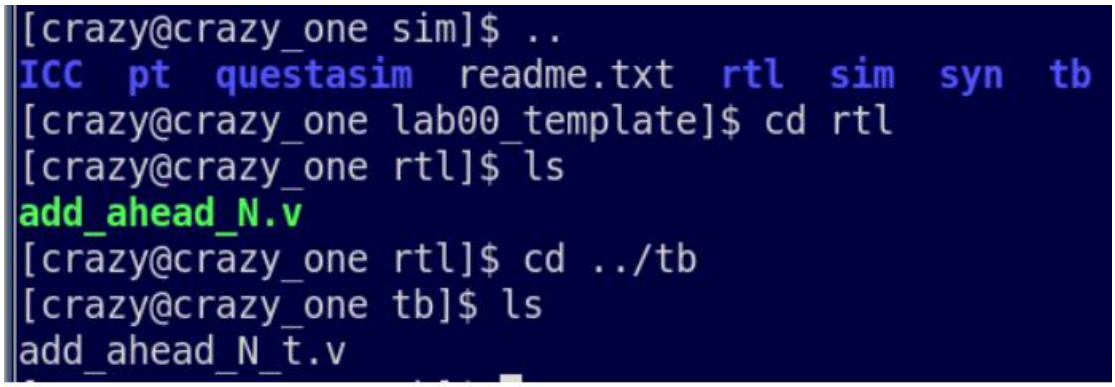
在 Linux 系统下，我们一般使用批处理文件 Makefile 来脚本化操作。先看一段脚本：

```
# Makefile For Verdi

all: compile simulate

compile:
    vcs \
    -debug_all \                    #在完全模式下进行调试，可替换为-debug_access+all
    -fsdb +define+FSDB \            #生成fsdb文件，并对所有.v源文件进行宏定义
    -l com.log \                    #生成com.log文件用于查看编译日志
    -f verif.f                      #文件列表，将verif.f所列文件全部编译

simulate:
    ./simv -l sim.log               #生成sim.log文件用于查看仿真日志

dve:
    dve \
    -vpd *.vpd &                    #启动Dve查看vpd类型的波形文件

verdi:
    verdi \
    -f verif.f \                    #用Verdi加载verif.f所列全部源文件
    -nologo \
    -ssf *.fsdb &                   #启动Verdi查看fsdb类型的波形文件

clean:
    @rm -rf csrc DVEfiles simv simv.daidir ucli.key VCS*
    @rm -rf *.log *.vpd *.ddc *.svf *.SDF *Synth *Netlist*
    @rm -rf alib-52 work
    @rm -rf *.conf *.rc *.fsdb verdiLog
```

1.2 源文件

以超前进位加法器的仿真为例，在文件夹（/rtl）与（/tb）中分别新建好 rtl 设计代码源文件与 testbench 测试文件。

如下图：



rtl 代码：

```
module add_ahead_N #(parameter SIZE=8)
(
    input  wire           cin,
    input  wire [SIZE-1:0] a,b,
    output wire [SIZE-1:0] sum,
    output wire           cout
);

wire [SIZE-1:0] G,P;
wire [SIZE:0] C;

assign C[0]=cin;
assign cout=C[SIZE];

generate
genvar i;
for(i=0;i<SIZE;i=i+1) begin:AHEAD
    assign G[i]=a[i]&b[i];
    assign P[i]=a[i]|b[i];
    assign C[i+1]=G[i]|(P[i]&C[i]);
    assign sum[i]=G[i]^P[i]^C[i];
end
endgenerate

endmodule
```

testbench 代码：

```
`timescale 1ns/1ns
module add_ahead_N_t;

parameter SIZE=8;
reg  [SIZE-1:0] a,b;
reg             cin;
wire [SIZE-1:0] sum;
wire            cout;

//add_ahead_N #(SIZE) AHN (cin,a,b,sum,cout);
add_ahead_N   #(SIZE) AHN (cin,a,b,sum,cout);

initial begin
a=0;
b=0;
cin=0;
end

initial begin
#23
cin=0;
a=24;
b=7;

#32
```

```
cin=1;
a=42;
b=12;

#100
$finish;
end

initial begin
    $vcdpluson();
end

`ifdef FSDB
initial begin
    $fsdbDumpfile("add_ahead_N_t.fsdb");
    $fsdbDumpvars;
end
`endif

endmodule

```

1.3 fsdb 文件

fsdb 文件是一种波形文件，只能用 Verdi 软件打开。要想得到 fsdb 文件，除了要将-fsdb 命令添加到前文提到的 Makefile 文件中外，还要在 testbench 文件中添加如下代码：

```
`ifdef FSDB
initial begin
    $fsdbDumpfile("add_ahead_N_t.fsdb");    //add_ahead_N_t为测试文件名称，可替换为其它名称
    $fsdbDumpvars;
end
`endif
```

2、软件使用教程

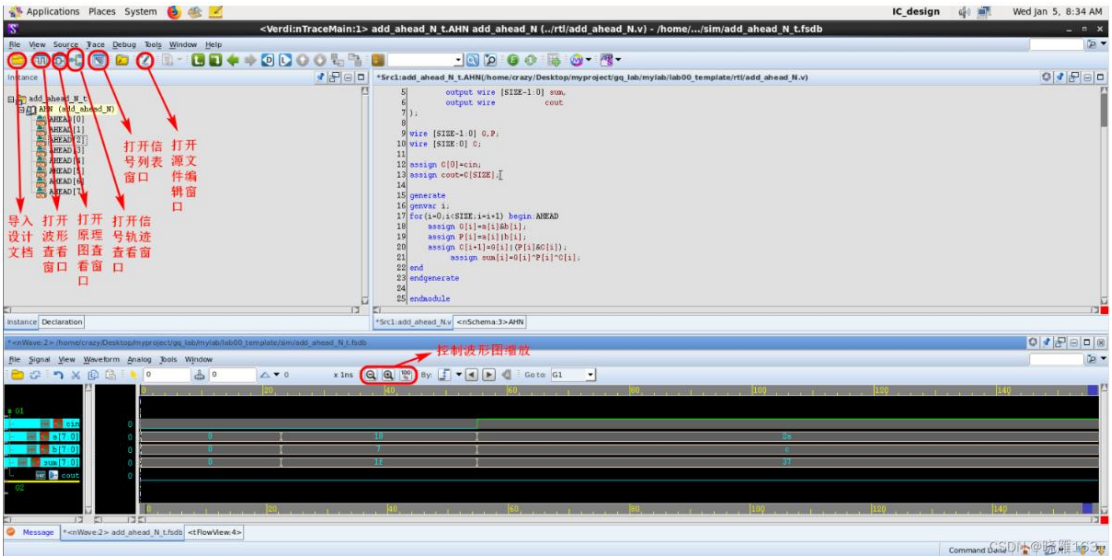
2.1 启动 Verdi

Verdi 可以在 Terminal 中通过命令行输入 verdi 直接启动，然后在图形界面下添加所需要的源文件、fsdb 文件。下面通过脚本的方式启动 Verdi。

在文件夹（/sim）中，打开 Terminal 终端，输入如下命令：

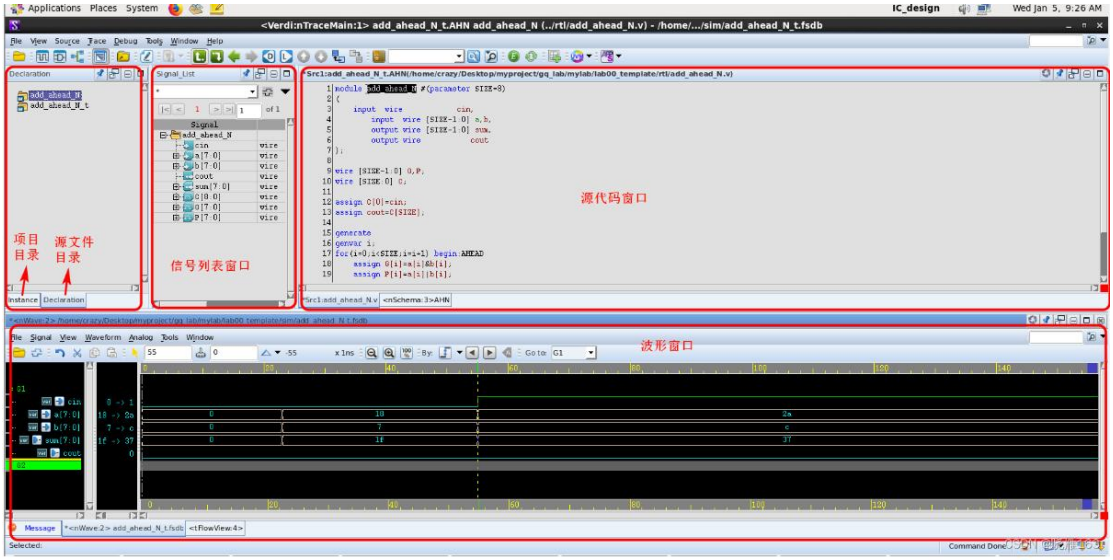
make

这时会弹出 Verdi 的图形化窗口，如下图：



2.2 Verdi 快捷方式

先来简单认识一下 Verdi 的窗口：



快捷键要在对应的窗口中使用，使用之前一定要用鼠标左键点一下窗口空白处！

源代码窗口：

x #显示信号的变化

ctrl+w #添加信号到波形

波形窗口：

f #将波形图 100%显示

z #缩小波形

shift+z #放大波形

h #开关显示信号结构

Del #删除信号

g #加载信号

鼠标操作：

拖拽信号 #左键点击选择一个信号，然后按住鼠标中键（滚轮）不松手，向其它窗口拖拽