

DC 实验指导书

陈虹，张吉霖

一、文件准备

1.1 建立工作目录

建立用来进行实验的项目目录（例如 testDC），命令为 `mkdir testDC`

1.2 准备 RTL 级源码

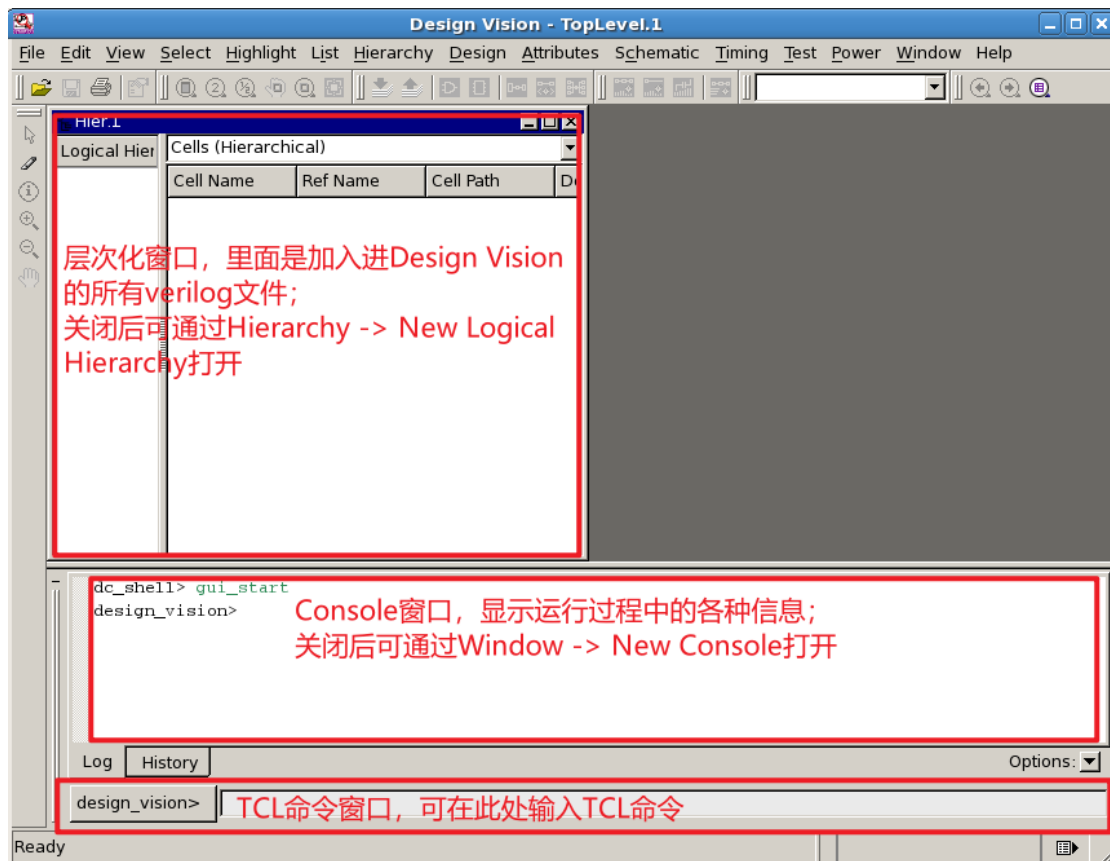
本次实验采用的是 counter.v。

```
`timescale 1ns/1ns
module counter(out,clk,rst);
    input rst,clk;
    output reg [3:0] out ;
    always @(negedge rst or posedge clk)
    begin
        if ( !rst )
            out <= 0;
        else
            out <= out + 1;
        end
    endmodule
```

二、DC 综合

2.1 启动 Design Vision

design compiler 有四种运行方式，我们使用支持 TCL 的图形界面 Design Vision，启动命令为：`design_vision &`

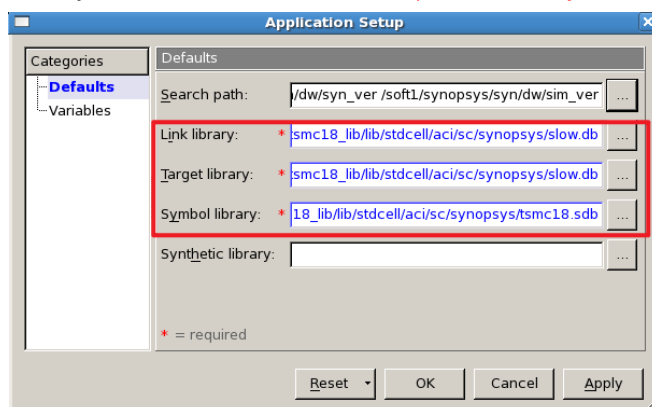


2.2 设置 Library

本次实验中使用的是 TSMC 180nm 的库，库处在 /data2/class/chenh/ic35/tsmc18_lib 文件夹下，Link library、Target Library 和 Symbol Library 的目录分别如下

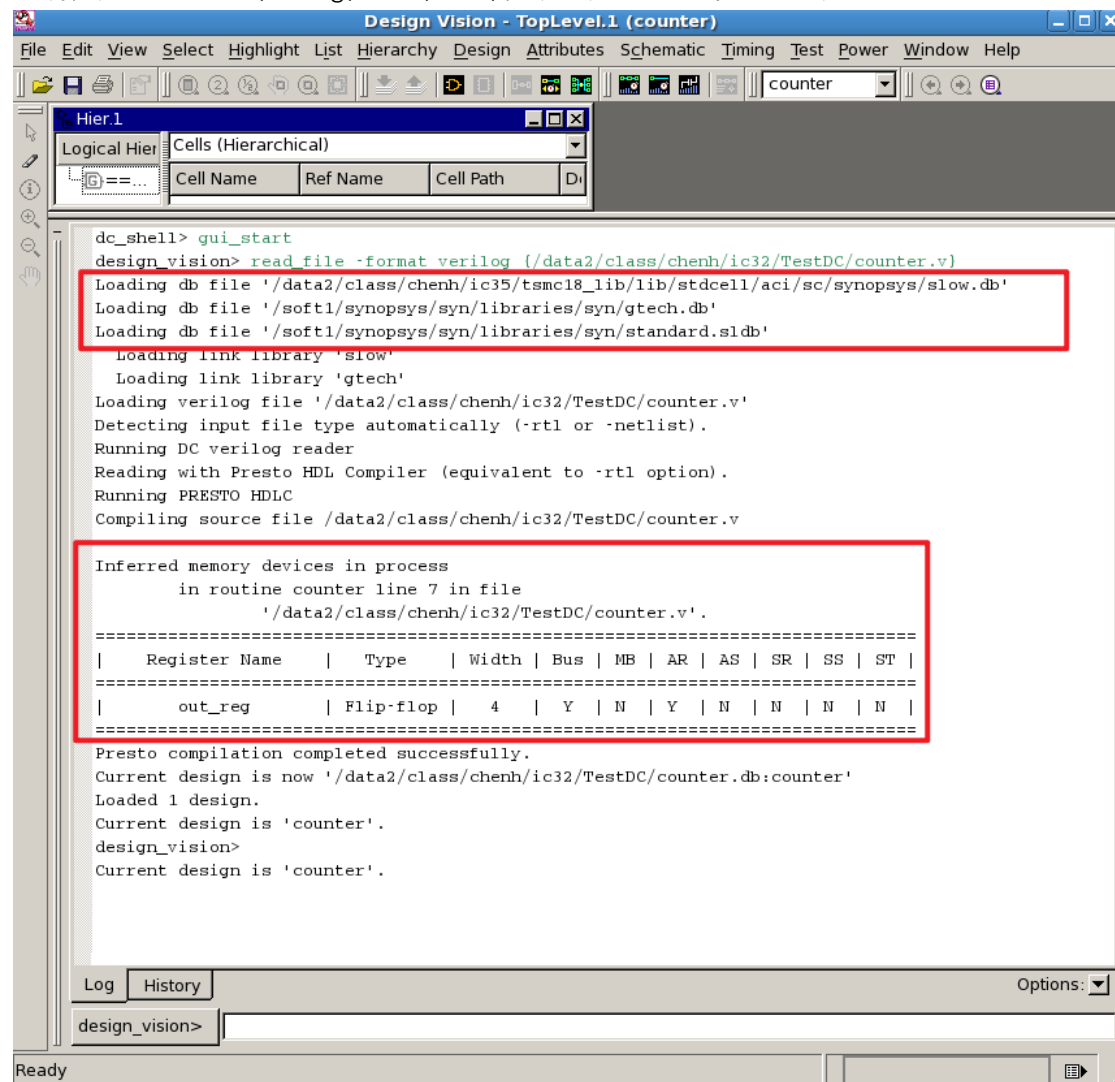
Link library: /data2/class/chenh/chenh45/class_lib/lib/stdcell/aci/sc/synopsys/slow.db
 Target library: /data2/class/chenh/chenh45/class_lib/lib/stdcell/aci/sc/synopsys/slow.db
 Symbol library: /data2/class/chenh/chenh45/class_lib/lib/stdcell/aci/sc/symbols/synopsys/tsmc18.sdb

在 Design Vision 中选择“File -> Setup...”菜单，确认 Link library、Target Library 和 Symbol Library 已经如下图所示设置好。(注意：要将 your_library.db 给删掉)



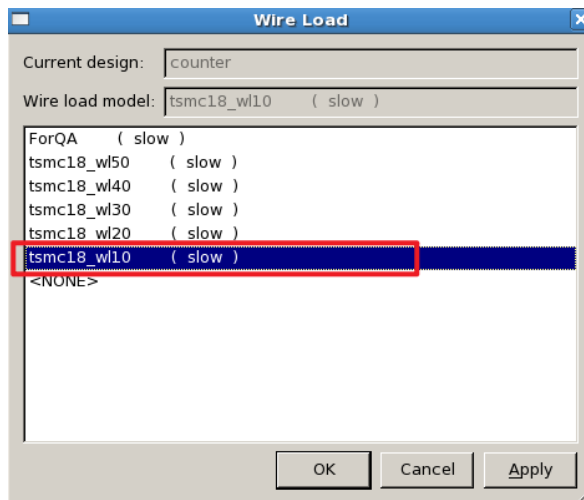
2.3 读入 RTL 代码

选择“File-> Read ...”菜单，选择需要综合的 RTL 代码文件 (counter.v)。读入 RTL 代码后，在 Console 窗口的 Log 页里显示了对 RTL 代码的分析结果，要仔细检查，库的设置是否正确，是否有警告 (warning) 或 (error)，以及是否和设计意图一致。



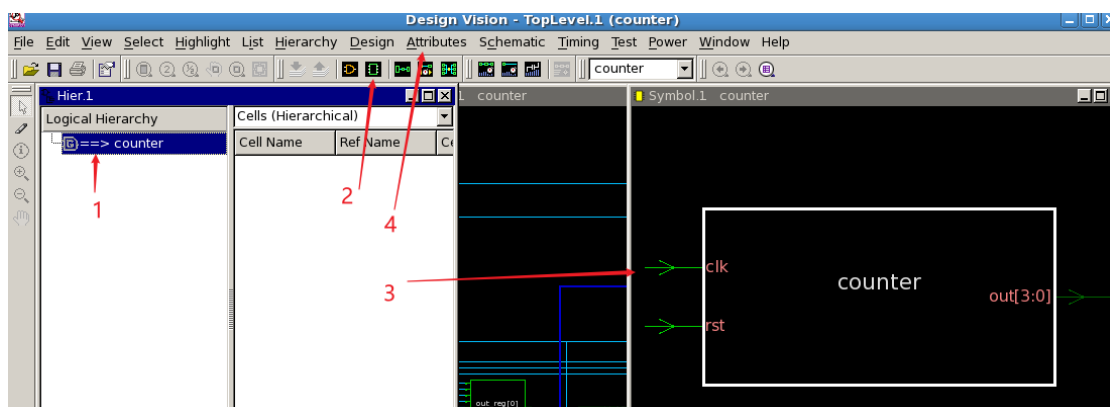
2.4 设置互连线模型 – Wire Load

选择“Attributes -> Operating Environment -> Wire Load”菜单，根据设计的等效门数的范围选择互连线模型。本实验的规模很小，所以选择最小的互连线模型。



2.5 设置时钟约束

- 在层次化浏览窗口选择顶层设计（counter）
- 打开符号图（symbol）或原理图（schematic）
- 选择时钟管腿或端口（clk），选中后相应的管腿会变成白色。
- 选择“Attributes -> Specify Clock”菜单



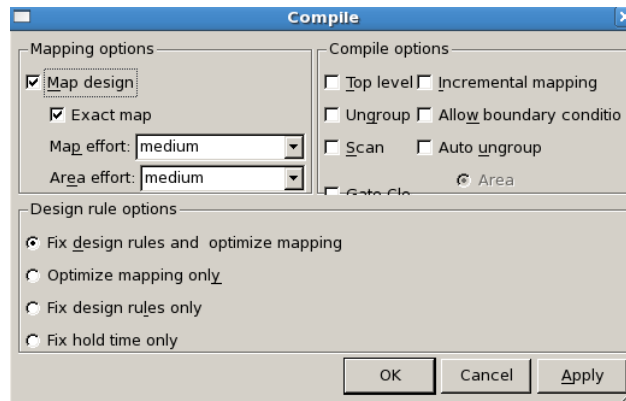
在弹出的窗口里设置时钟的参数

- 随便给时钟起一个名字（clock name），例如 CK
- 确定 Port name 处是选择的时钟管腿的名字（灰白色显示）
- 设置时钟周期（Period），单位是 ns，例如 10 ns
- 时钟上升沿（Rising）和下降沿（Falling）的相对时间，一般上升沿相对时间为 0，下降沿相对时间为 T/2，即时钟信号是占空比 50% 的方波信号。
- 选择“Don't touch network”，使综合工具不对时钟信号插入缓冲树。

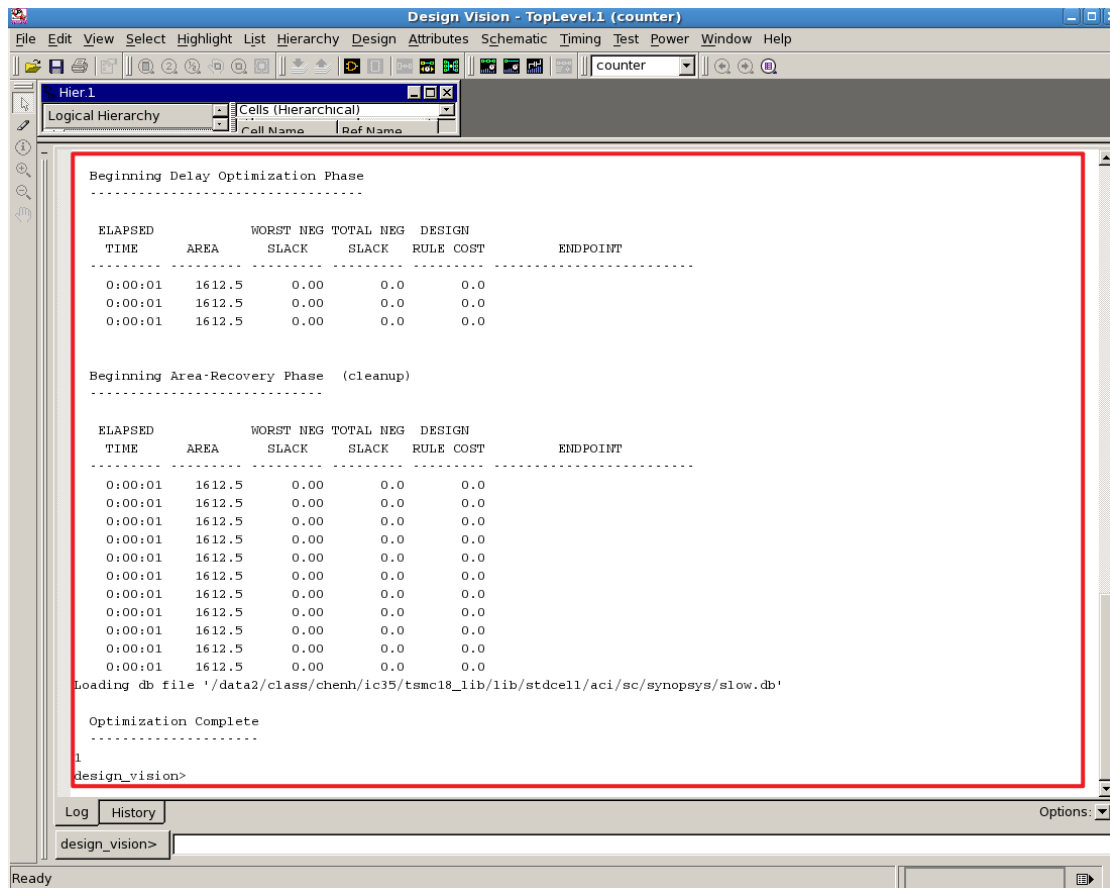


2.6 综合

选择“Design -> Compile Design”菜单，保持默认点 OK。



在控制台窗口里会显示综合的结果，包括面积等信息



2.7 检查综合结果 – Area

选择“Design -> Report Area”菜单，在控制台窗口可以看到以下报告：

```
*****
Report : area
Design : counter
Version: D-2010.03
Date   : Tue Nov 10 18:46:44 2020
*****

Information: Updating design information... (UID-85)
Library(s) Used:

    slow (File: /data2/class/chenh/ic35/tsmc18_lib/lib/stdcell/aci/sc/syno

Number of ports:          6
Number of nets:          12
Number of cells:          9
Number of references:     6

Combinational area:      103.118402
Noncombinational area:   296.049599
Net Interconnect area:   1213.339417

Total cell area:         399.168000
Total area:              1612.507417

***** End Of Report *****
```

组合逻辑面积

时序逻辑面积

估计的互连线面积

单元（器件）的总面积

总面积

2.8 检查综合结果 – Timing

选择“Timing -> Report Timing Paths”菜单，在弹出的窗口点击“ok”。

Startpoint: out_reg[0] (rising edge-triggered flip-flop clocked by CK)
Endpoint: out_reg[3] (rising edge-triggered flip-flop clocked by CK)
Path Group: CK
Path Type: max

Des/Clust/Port	Wire Load Model	Library	
counter	tsmc18_wl10	slow	

Point	Incr	Path
clock CK (rise edge)	0.00	0.00
clock network delay (ideal)	0.00	0.00
out_reg [0]/CK (JKFFRXL)	0.00	0.00 r
out_reg [0]/Q (JKFFRXL)	1.17	1.17 r
U10/Y (NAND2X1)	0.28	1.46 f
U14/Y (NOR2BX1)	0.31	1.76 r
U13/Y (XOR2X1)	0.33	2.10 f
out_reg [3]/D (DFFRHQX1)	0.00	2.10 f
data arrival time		2.10
clock CK (rise edge)	10.00	10.00
clock network delay (ideal)	0.00	10.00
out_reg [3]/CK (DFFRHQX1)	0.00	10.00 r
library setup time	-0.37	9.63
data required time		9.63
data required time		9.63
data arrival time		-2.10
slack (MET)		7.53

器件的延时

所在路径的总延时

关键路径经过的器件

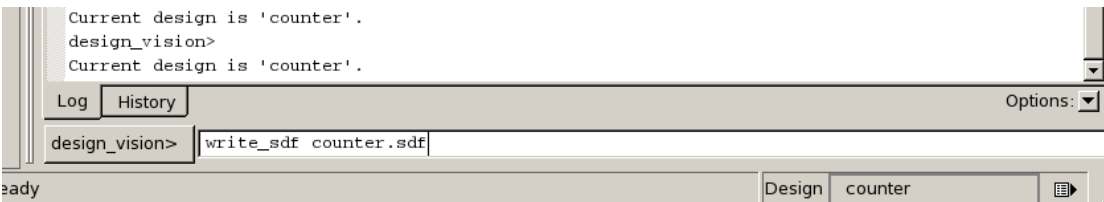
约束的时钟周期

允许的最大延时

裕量

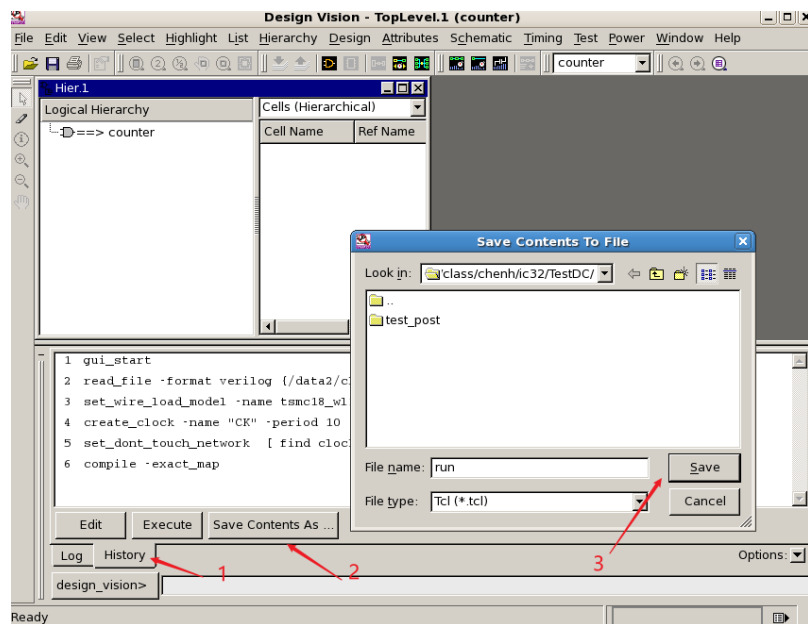
2.9 保存综合结果

- 选择“File -> Save”菜单，以 DDC 格式保存综合后的结果（counter.ddc）。以后如果需要再次分析综合结果，可以直接读入相应的 DDC 文件。
- 选择“File -> Save As”，以 Verilog 形式输出电路网表，用于门级仿真和后端布局布线。Verilog 文件的缺省后缀是“.v”，文件类型选“Auto”或“Verilog”，例如：counter_post.v，注意文件名不要和 RTL 代码重复。
- 在 DC 中，使用 write_sdf 命令导出时序描述文件。



2.10 以 TCL 方式运行 DC

在每一步图形操作后，控制台窗口的 History 记录下和图形操作对应的 TCL 命令。另外，在运行 Design Vision 的目录下会产生一个 command.log 日志文件，里面也记录了这些 TCL 命令。把 command.log 文件拷贝成 run.tcl。如下图所示。



删除无关的记录，只保留以下 TCL 命令，并在最后增加一行“quit”：

```
set target_library { /data2/class/chenh/chenh45/class_lib/lib/stdcell/aci/sc/synopsys/slow.db }
set link_library { * /data2/class/chenh/chenh45/class_lib/lib/stdcell/aci/sc/synopsys/slow.db }
set symbol_library { /data2/class/chenh/chenh45/class_lib/lib/stdcell/aci/sc/symbols/synopsys/tsmc18.sdb }

read_file -format verilog {src/counter.v}
set_wire_load_model -name tsmc18_wl10 -library slow
create_clock -name "CK" -period 10 -waveform { 0 5 } { clk }
set_dont_touch_network [ find clock CK ]
compile -exact_map
write -hierarchy -format ddc
write -hierarchy -format verilog -output counter_post.v
write_sdf counter.sdf
quit
```

当需要对 RTL 代码重新综合时，就可以在 LINUX 终端窗口里(工作目录下)用以下命令以批处理的方式自动运行了，命令

```
dc_shell-t -f run.tcl
```

注意: target_library、link_library 以及 symbol_library 与后面的括号间存在空格!

三、门级仿真

3.1 文件准备

对综合后输出的网单文件（例如 counter_post.v）进行门级仿真，使用在功能仿真中使用的 testbench（例如 tb_count.v）。在 tb_count.v 中添加 \$sdf_annotate("counter.sdf", uut);，具体如下：


```

`timescale 1ns/1ps

module tb_count;

reg      rst, clk;
wire     [3:0] dout;

counter  uut(dout, clk, rst);

initial begin
$sdf_annotate("counter.sdf", uut);
clk = 0;
rst=1;
#100;
rst=0;
#100;
rst=1;
#1000;
$finish;
end

always #5 clk = ~clk;

initial begin
$fsdbDumpfile("counter.fsdb");
$fsdbDumpvars(0, tb_count);
$fsdbDumpon;
end

endmodule

```

另外，还需要把标准单位的仿真库文件 tsmc18.v（已上传到网络学堂）拷贝到当前项目目录里。确认当前目录，并检查所需文件是否都存在工作目录下。

3.2 修改 filelist.f 以及 Makefile 文件

修改 filelist.f 文件，将综合得到的 counter_post.v，tsmc18.v，以及 tb_cnt.v 添加进去。

```

//Source file
./src/counter_post.v
./src/tsmc18.v

//Testbench
./src/tb_count.v

```

同时修改 Makefile 文件，将加载的 fsdb 文件指向 counter.fsdb。

```

.PHONY: com verdi clean

#compile command
VCS = vcs +v2k -timescale=1ns/1ps \
      -full64 \
      -R \
      -debug_access+all \
      -f filelist.f \
      -l vcs.log

VERDI=Verdi-SX -f filelist.f \
               -ssf counter.fsd \
               -l verdi.log

#start compile and simulate
com:
    $(VCS)

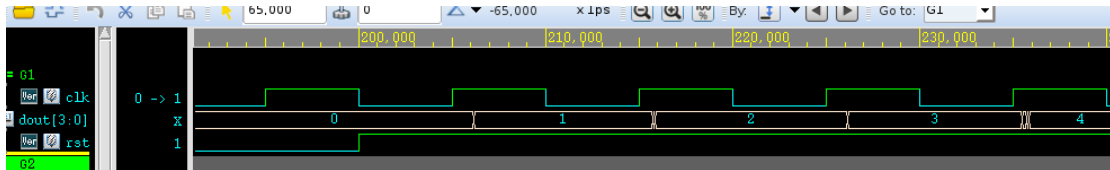
#run verdi
verdi:
    $(VERDI) &

#clean
clean:
    rm -rf ./Verdi-SXLog ./dff ./csrc *.daid:
    *race.out* *.rc *.fsdb *.vpd *.log *.conf *.dat *.c

```

3.3 运行 VCS 以及 Verdi 查看波形

运行 VCS 以及 Verdi 查看波形，将信号添加入波形窗口，在时钟与信号间看到延时则意味着成功加载 sdf 文件，门级仿真成功。



3.4 VCS 针对 sdf 的命令

下列命令添加在 VCS = vcs +v2k ... 这一系列语句的后面，从而对 sdf 反标做出更多的调整与约束。

- +mindelays 器件延时使用 sdf 文件中的最小值(sdf 文件中的时序(min:typ:max))
- +maxdelays 器件延时使用 sdf 文件中的最大值
- +neg_tchk 负延迟检查，P&R 后仿需要加上此选项
- -notimingcheck 忽略仿真中的时序检查，会忽略 specify 语句中的检查，慎用。