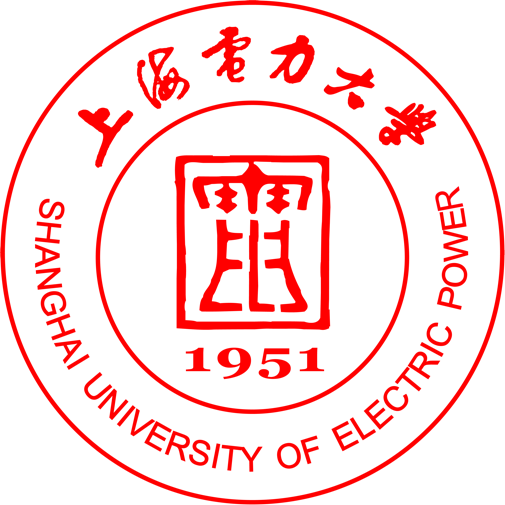
**上 海 电 力 大 学**

《数字集成电路设计与分析》实验报告



**实验题目：**  DC综合实例实验

**专 业：**

**班 级**  **学号**

**姓 名**

**时 间**  2023-11-08

1. 实验目的

基于给定的原理图及其说明定义 DC 环境属性

将属性应用于设计

验证应用的属性

1. 实验任务及要求

从 Lab3/ib.rpt 文件当中查看负载电容单位

复制lab3/scripts/MY\_DESIGN.con到lab4/scripts/MY\_DESIGN.con

将 4-4 表格定义的说明添加到 MY\_DESIGN.con 当中

验证并修改约束文件语法

调用DC

读取rtl/MY\_DESIGN.v文件

写出约束文件，并与 solutions/MY\_DESIGN.wscr 对比

退出 DC

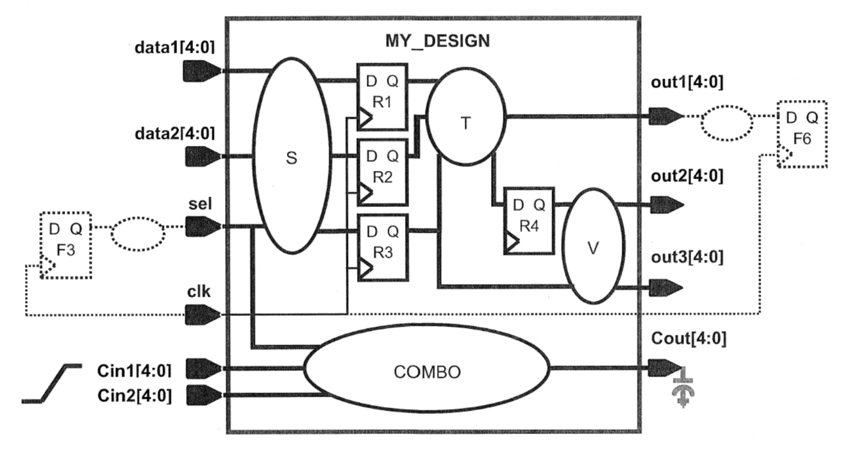


Figure39 设计原理图

1. 关于约束文件

**CLEAN UP**

reset\_design

删除任何现有的约束和属性

**CLOCK DEFINITION**

create\_clock -period 3.0 [get\_ports clk]

333Mhz 时钟是 3.0ns 周期。

set\_clock\_latency -source -max 0.7 [get\_clocks clk]

外部时钟源延迟为 700ps 或 0.7ns

set\_clock\_uncertainty -setup 0.15 [get\_clocks clk]

如果延迟启动 (+30ps) 并提前捕获 (-30ps)，则寄存器时钟引脚的 +/-30ps 内部时钟延迟变化会导致 60ps 最坏情况的偏差或不确定性； 添加 40ps 由于抖动和 50ps 用于设置余量；

这相当于 150ps 或 0.15 ns 的总不确定度。

set\_input\_transition 0.12 [get\_ports Cin\*]

端口 Cin 是芯片级输入，输入转换为 120ps 或 0.12 ns

**INPUT TIMING**

set\_input\_delay -max 0.45 -clock clk [get\_ports data\*]

端口 detal 和 data2 上的最大“输入延迟”（外部）为：

clock period - clock uncertainty - delay of S - register setup time =

3.0 - 0.15 - 2.2 - 0.2 = 0.45 ns

set\_input\_delay max 0.4 -clock clk [get\_ports sel]

最晚到达sel端口的时间是1.4ns（绝对时间）。 总时钟插入延迟或外部寄存器的延迟为 700ps + 300ps 或 1.0ns。 因此，端口上的相对输入延迟为 1.4 -1.0 = 0.4ns

**OUTPUT TIMING**

set\_output\_delay -max 0.5 -clock clk [get\_ports out1]

The output delay at port out1 is 420ps + 80ps = 500ps or 0.5ns

set\_output\_delay -max 2.04 -clock clk [get\_ports out2]

out2 的内部延迟为 0.81ns。 外部捕获时钟沿发生在启动沿后 3ns，减去 0.15ns 的不确定性，即启动后 2.85ns。 为了将内部延迟限制为 0.81ns，输出延迟必须限制为 2.85ns - 0.81ns = 2.04ns。

set\_output\_delay max 0.4 -clock clk [get\_ports out3]

相对于捕获寄存器的时钟，端口 3 上的建立时间要求为 400ps 或 0.4ns。

根据定义，这是“set\_output\_delay”值。

**COMBINATIONAL LOGIC TIMING**

set\_output\_delay -max 0.1 -clock clk [get\_ports Cout]

set\_input\_delay -max 0.3 -clock clk [get\_ports Cin\*]

组合逻辑的最大延迟为 2.45ns。 这可以通过假装输入端口 Cin1 和 Cin2 上有启动寄存器并在输出端口 Cout 上有捕获寄存器并应用相应的输入和输出延迟来限制。 外部输入和输出延迟值之和必须等于时钟周期减去时钟不确定性再减去最大组合延迟 = 3ns - 0.15ns - 2.45ns = 0.4ns。

这意味着输入和输出延迟值可以分别为0.4和0.0、或0.2和0.2、或0.1和0.3等。

**ENVIRONMENTAL ATTRIBUTES**

set\_driving\_cell -lib\_cell bufbd1 -library cb13fs120\_tsmc\_max \

[remove\_from\_collection [all\_inputs] [get\_ports "clk Cin\*"]]

除了 clk 和 Cin 之外，所有输入端口都由 bufbd1 缓冲器驱动

set\_load -max [expr {2 \* [load\_of cb13fs120\_tsmc\_max/bufbd7/I]}] [get\_ports out\*]

所有输出，除了 Cout，驱动 2x bufbd7 负载。

set\_load 0.025 [get\_ports Cout\*]

Cout 驱动 25fF，或 .025 pF

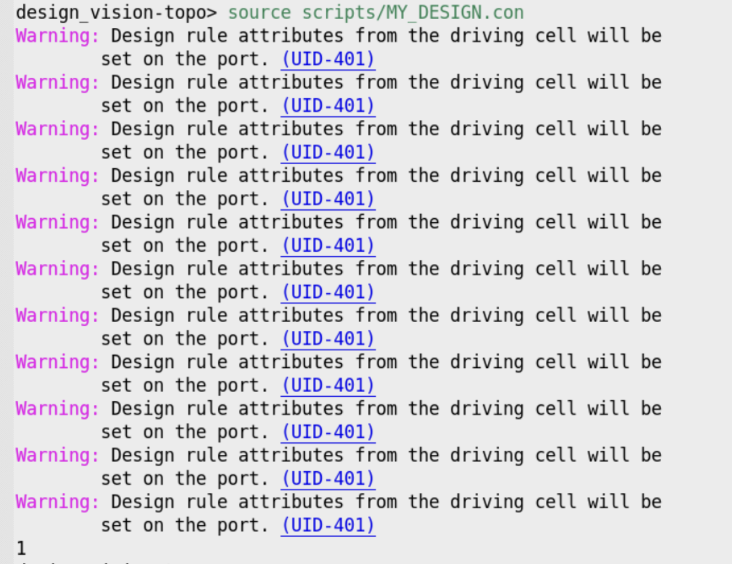
1. 实验内容及步骤

1 创建scrips目录，创建MY\_DESIGN.con约束文件。

2 读取，链接，检查rtl/MY\_DESIGN.v文件。

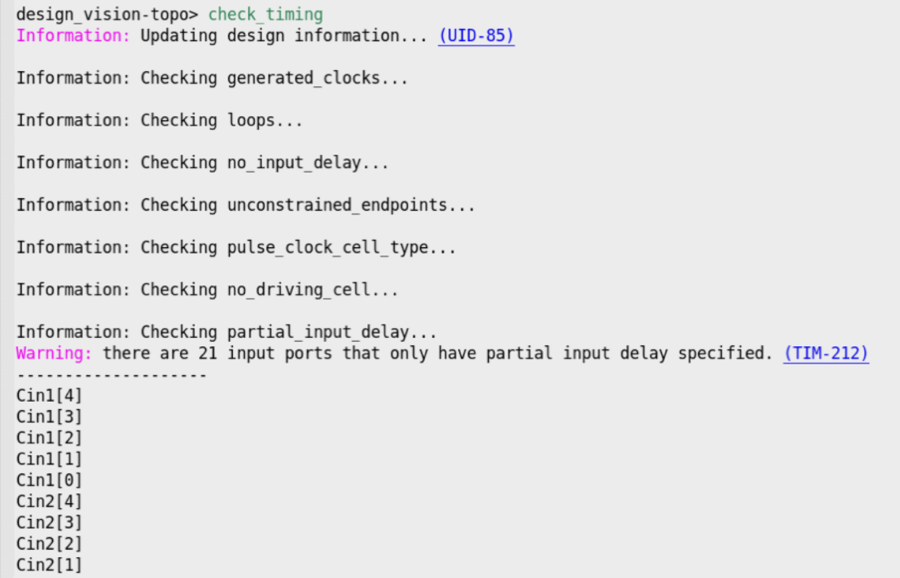
3 输入以下命令，调用约束文件。

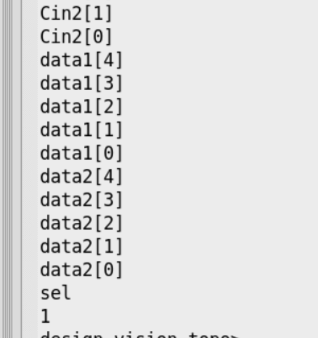
source scripts/MY\_DESIGN.con



4 输入以下命令，检验关键约束是否存在缺失或冲突

check\_timing



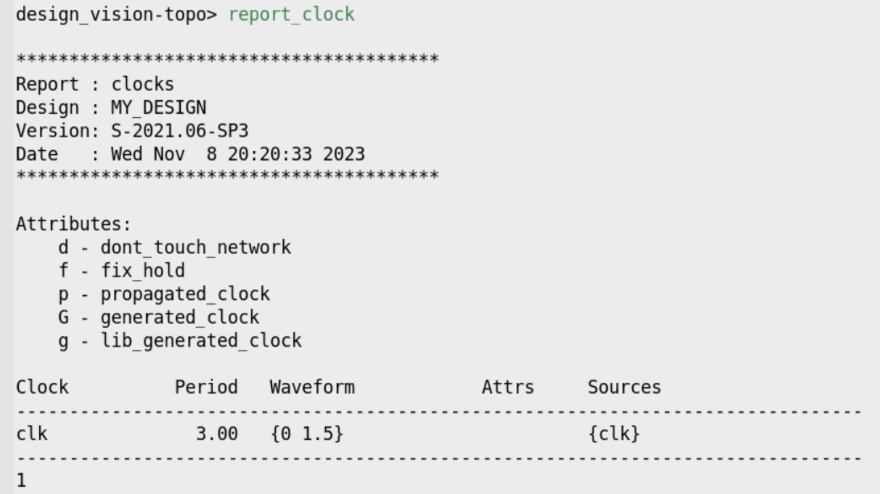


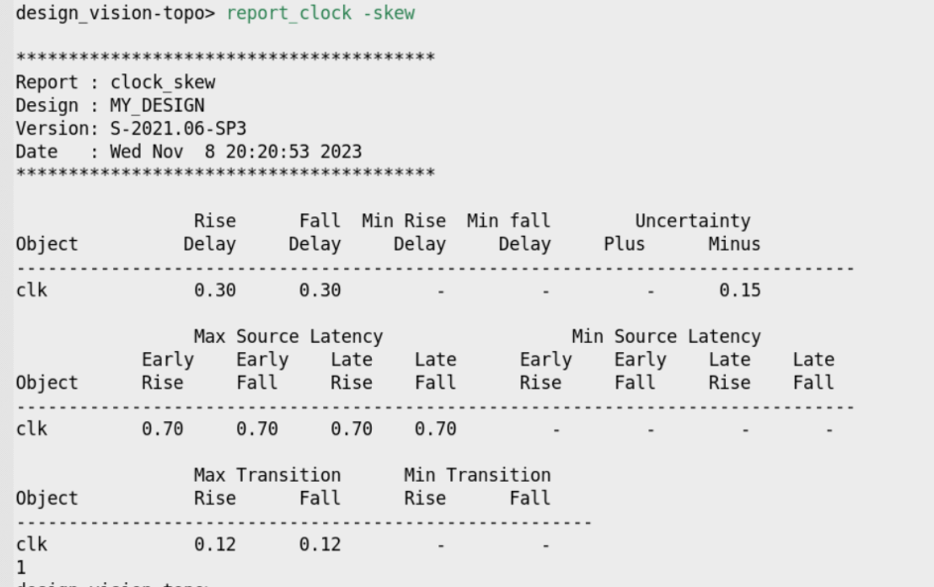
5 输入以下命令，验证时钟和端口约束

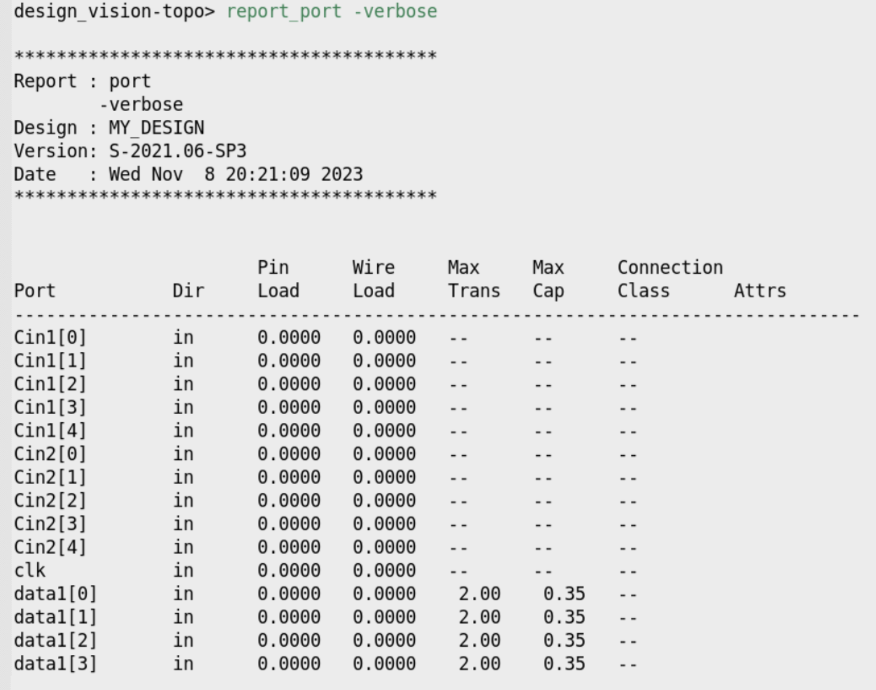
report\_clock

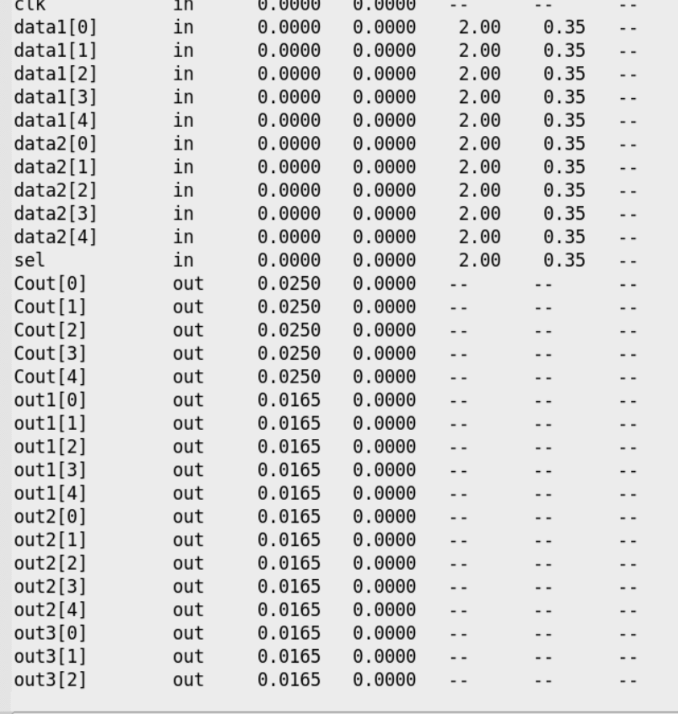
report\_clock -skew

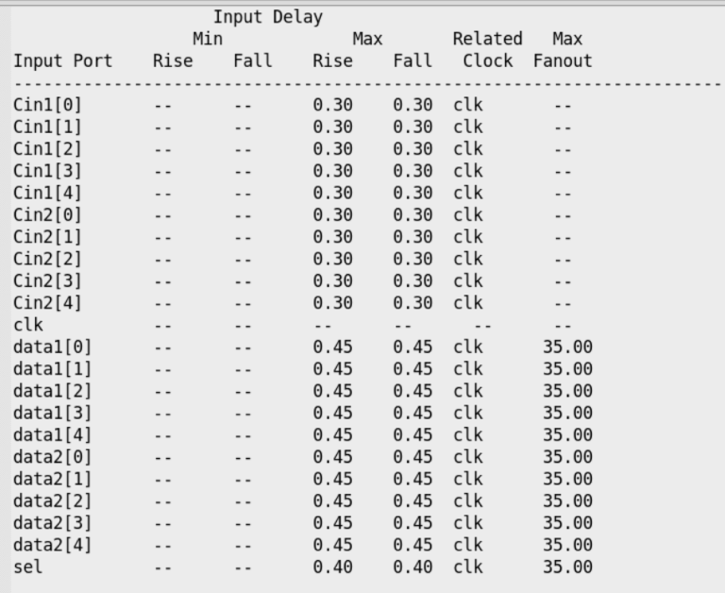
report\_port -pverbose

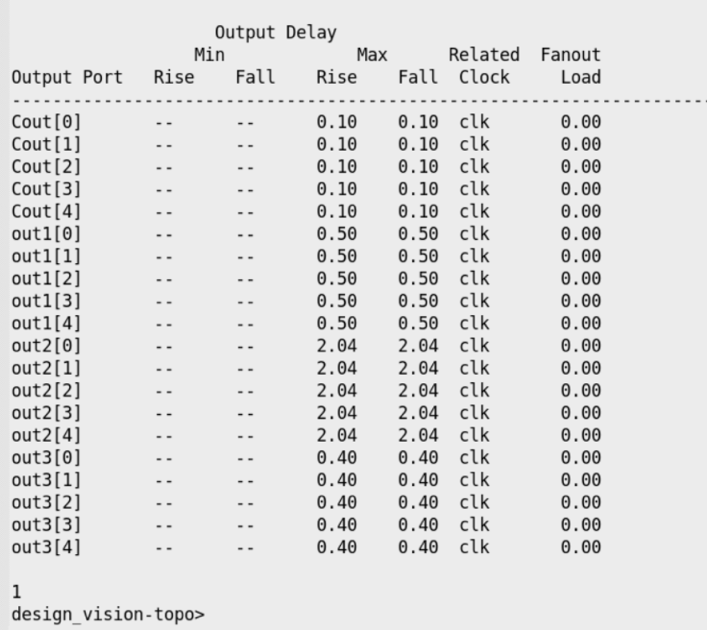










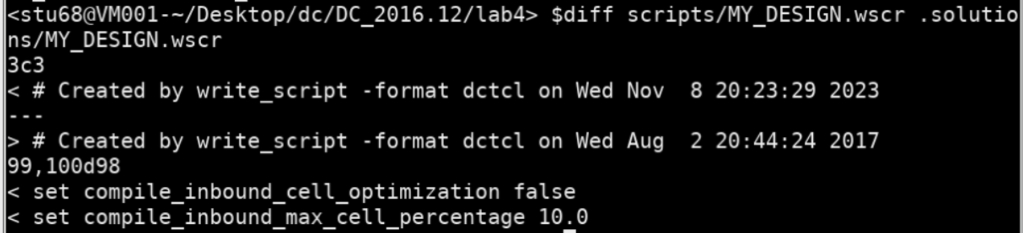


6 输入以下命令，以展开形式将约束导出

write\_script -out scripts/MY\_DESIGN.wscr

7 在LUNIX命令窗口输入以下命令，比对展开后约束文件同原始文件区别

diff scripts/MY\_DESIGN.wscr .solutions/MY\_DESIGN.wscr



8 输入以下命令，保存 ddc 文件

write -format ddc -hier -out unmapped/MY\_DESIGN.ddc



9 退出DC

1. 实验总结