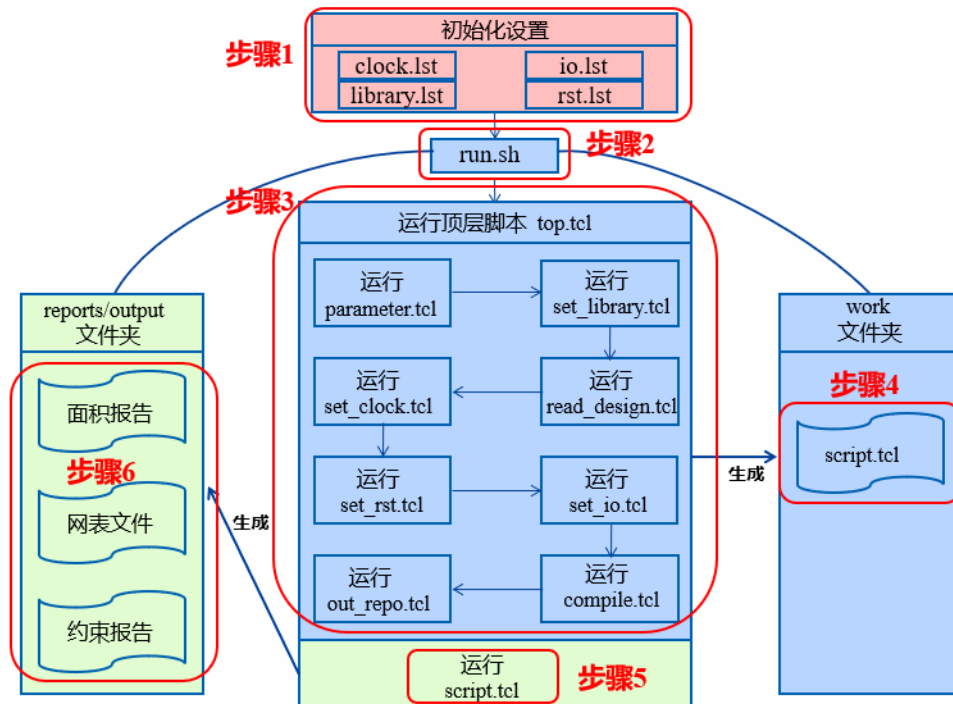


# 基于 SYNOPSIS DC 逻辑综合约束 自动化生成框架介绍

## 一. 脚本介绍

### 1. 脚本框架

脚本框架如下图所示



其中 common 文件夹下脚本用于生成逻辑综合脚本 script.tcl, setup 文件夹用于指定设计的时钟信号, 复位信号, 顶层设计, 工艺库等, 后面将详细介绍 setup 设置的格式, 在本教程中 aes\_core 为参考的 rtl 设计。

### 2. 使用流程

参考 readme.txt

```
1:复制rtl代码文件夹到rtl_code文件夹下
2:修改setup中各个初始化文件的设置
3:在SYN_FLOW文件夹下, 打开linux系统的terminal命令行, 输入source run.sh "X"
  (X为希望指定的DC_log的名字)
```

### 3. 注意事项

如果需要只生成 scripts.tcl, 不运行 DC, 则在 SYN\_FLOW 文件夹下, 打开 linux 系统的 terminal 命令行输入 source ./common/top.tcl

## 二. Setup 设置

### 1. clock.lst 设置

clock.lst 格式如下图所示

```
#####
#                CLOCK
#####
#
#ClockName Period Rise Fall ClockPort
#-----
clk 10 0 5 clk
```

时钟的设置从分割线下面一排开始，从左往右分别设置时钟名字，周期，上升沿，下降沿，时钟 PORT，在前两行中约束时钟端口为 PORT;如图最后一行中，可以约束时钟端口为设计内部一个模块的 pin。

此时生成的 script.tcl 中 clock 部分如下图所示

```
#####CLOCK#####
remove_driving_cell [get_ports clk]
set_drive 0 [get_ports clk]
create_clock -name clk [get_ports clk] -period 10 -waveform [list 0 5]
set_dont_touch_network clk
set_ideal_network -no_propagate [get_ports clk]
#####SKEW & LATENCY#####
set_clock_uncertainty 0.5 [get_clocks clk]
set_clock_latency -source -max 1.0 [get_clocks clk]
set_clock_latency -max 1.0 [get_clocks clk]
set_clock_transition -max 0.1 [get_clocks clk]
```

其中有关 skew, latency 以及 transition 数值设置的部分如下图所示，可以根据设计的具体需求做出更改

```
set CLK_SKEW [expr $PRD*0.05]
set CLK_SOURCE_LATENCY [expr $PRD*0.1]
set CLK_NETWORK_LATENCY [expr $PRD*0.1]
set CLK_TRAN [expr $PRD*0.01]
```

同时，当设计时钟大于两个时，clock 约束部分会生成有关 false path 的约束，如下图所示

```
#####FALSE_PATH#####
set_false_path -from [get_clocks clk1] -to [get_clocks clk2]
set_false_path -from [get_clocks clk2] -to [get_clocks clk1]
```

## 2. io.lst 设置

### 1) 设置具体 IO 口的约束

如下图所示

```
#####
#                IO
#####
#
#I/O I/O_Port CKNAME max_Delay min_Delay max_Delay_o min_Delay_o
#-----
I key* clk 2 0
O text_out clk 2 0
```

此时需要从左往右依次需要指定端口的方向，端口 PORT 名称，时钟 PORT 名称，以及 maxdelay 和 mindelay，这种情况下只用设置 min\_Delay 以及 max\_Delay，max\_Delay\_o 和 min\_Delay\_o 不用设置，脚本会根据端口方向自动生成约束脚本

```
#####DELAY#####
set_input_delay 2 -max -clock [get_clocks clk] [get_ports key*]
set_input_delay 0 -min -clock [get_clocks clk] [get_ports key*]
set_output_delay 2 -max -clock [get_clocks clk] [get_ports text_out]
set_output_delay 0 -min -clock [get_clocks clk] [get_ports text_out]
```

### 2) 仅设置时钟以及 IO 延迟时间

此时将 I/O 方向，以及 I/O\_PORT 设置为 0，后续依次设置约束时钟名字，输入最小最大延迟，输出最小最大延迟。(注意：此时需要设置 4 个延时值，跟上一情况不同)

```
#####
#                               IO
#####
#
#I/O I/O_Port CKNAME max_Delay min_Delay max_Delay_o min_Delay_o
#-----
0 0 clk 6 0 4 0
```

参考该 io.lst，运行 source ./common/top.tcl 后，得到 DELAY 相关部分如下图所示，此时 output\_delay 应用到所有输出端口上，input\_delay 应用到除了 clk\_port 外的所有 input 端口上

```
#####DELAY#####
set_input_delay 6 -max -clock [get_clocks clk] [remove_from_collection [all_inputs] [get_ports "clk" ]]
set_input_delay 0 -min -clock [get_clocks clk] [remove_from_collection [all_inputs] [get_ports "clk" ]]
set_output_delay 4 -max -clock [get_clocks clk] [all_outputs]
set_output_delay 0 -min -clock [get_clocks clk] [all_outputs]
```

### 3. library.lst 设置

如下图所示，第一行设置工艺库的位置 root path，第二行设置用到的工艺库的名称 library name，第三行设置设计的顶层名称 top name。

```
#####
#                               library
#####
#root path
#librarName
#top name
#-----
../lib
sky130_fd_sc_hd_tt_025C_1v80.db
aes_cipher_top
```

### 4. rst.lst 设置

如下图所示，每行的第一个字符串为 rst 的名称，如果需要设置多个 rst，则需要换行继续写入另一个 rst

```
#####
#                               RST
#####
#
#RSTName
#-----
rst
```

### 5. compile 策略设置

如果需要修改 compile 的策略，可以打开 common 中的 compile.tcl，修改 puts \$fp\_write “compile\_ultra”这条代码，如下图所示，在演示中设置的是 compile\_ultra。

```
set outfile "./work/script.tcl"
set fp_write [open $outfile a]

puts $fp_write "#####COMPILE#####"

puts $fp_write "compile_ultra"
#puts $fp_write "compile_ultra -no_autoungroup -no_seq_output_inversion -no_boundary_optimization"
close $fp_write
```

如果需要添加综合相关的约束，可以在 compile.tcl 中输出 compile\_ultra 之前添加相关的约束，如下图所示。需要注意的是，当添加的约束涉及到\${},[],[]等符号时，需要在符号前面加上转义符\。

```
1 set outfile "./work/script.tcl"
2 set fp_write [open $outfile a]
3
4
5 puts $fp_write "#####COMPILE#####"
6 puts $fp_write "set_fix_multiple_port_nets -all -buffer_constants"
7 puts $fp_write "compile_ultra"
8 #puts $fp_write "compile_ultra -no_autoungroup -no_seq_output_inversion -no_boundary_optimization"
9 close $fp_write
```

### 三. 实例演示

#### 1. 准备工作

##### 1) 确定设计文件

这里实例使用的 rtl 设计为 aes\_core( 下载地址 : [https://github.com/mematrix/AES-FPGA/blob/master/aes\\_core/](https://github.com/mematrix/AES-FPGA/blob/master/aes_core/)), 将 rtl 代码复制到 rtl\_code 文件夹下, 如图所示

```
[eda@edacentos rtl_code]$ pwd
/home/eda/hwt/SYN_FLOW/rtl_code
[eda@edacentos rtl_code]$ ls
aes_core
```

#### 2. setup 设置

这里需要依次对 clock.lst, io.lst, library.lst, rst.lst 进行设置, 如下图所示

```
#####
#                                CLOCK
#####
#
#ClockName Period Rise Fall ClockPort
#-----
| clk 10 0 5 clk
#####
#                                IO
#####
#
#I/O I/O_Port CKNAME max_Delay min_Delay max_Delay_o min_Delay_o
#-----
| 0 0 clk 6 0 4 0
#####
#                                library
#####
#root path
#librarName
#top name
#-----
| ../lib
sky130_fd_sc_hd__tt_025C_1v80.db
aes_cipher_top
#####
#                                RST
#####
#
#RSTName
#-----
| rst
```

由此设置 source ../common/top.tcl 生成的 script.tcl 如下图所示

```

#####PRE#####
set_host_options -max_cores 16
set_svf ../output/aes_cipher_top_20240911_1100.svf
set_app_var search_path "../lib"
set_app_var target_library "sky130_fd_sc_hd_tt_025C_1v80.db "
set_app_var link_library "* sky130_fd_sc_hd_tt_025C_1v80.db"
set_top aes_cipher_top
#####READ_DESIGN#####
analyze -format verilog /home/eda/hwt/SYN_FLOW/rtl_code/aes_core/timescale.v
analyze -format verilog /home/eda/hwt/SYN_FLOW/rtl_code/aes_core/aes_sbox.v
analyze -format verilog /home/eda/hwt/SYN_FLOW/rtl_code/aes_core/aes_rcon.v
analyze -format verilog /home/eda/hwt/SYN_FLOW/rtl_code/aes_core/aes_key_expand_128.v
analyze -format verilog /home/eda/hwt/SYN_FLOW/rtl_code/aes_core/aes_cipher_top.v
elaborate aes_cipher_top
current_design aes_cipher_top
uniquify -force
check_design
link
write_file -format ddc -hierarchy -output ../output/aes_cipher_top_20240911_1100_link.ddc
#####CLOCK#####
remove_driving_cell [get_ports clk]
set_drive 0 [get_ports clk]
create_clock -name clk [get_ports clk] -period 10 -waveform [list 0 5]
set_dont_touch_network clk
set_ideal_network -no_propagate [get_ports clk]
#####SKEW & LATENCY#####
set_clock_uncertainty 0.5 [get_clocks clk]
set_clock_latency -source -max 1.0 [get_clocks clk]
set_clock_latency -max 1.0 [get_clocks clk]
set_clock_transition -max 0.1 [get_clocks clk]
#####SET_RST#####
set_dont_touch_network [get_ports rst]
set_false_path -from [get_ports rst]
set_ideal_network -no_propagate [get_ports rst]
set_drive 0 [get_ports rst]
#####DELAY#####
set_input_delay 6 -max -clock [get_clocks clk] [remove_from_collection [all_inputs] [get_ports "clk" ]]
set_input_delay 0 -min -clock [get_clocks clk] [remove_from_collection [all_inputs] [get_ports "clk" ]]
set_output_delay 4 -max -clock [get_clocks clk] [all_outputs]
set_output_delay 0 -min -clock [get_clocks clk] [all_outputs]

set_max_fanout 32 [current_design]
#####COMPILE#####
compile_ultra
#####output#####
write -format verilog -hierarchy -output ../output/aes_cipher_top_20240910_1612.v
write_sdc ../output/aes_cipher_top_20240910_1612.sdc
write_file -format ddc -hierarchy -output ../output/aes_cipher_top_20240910_1612_compile.ddc
write_sdf ../output/aes_cipher_top_20240910_1612.sdf
set_svf -off
#####report#####
report_area -nosplit -hierarchy > ../report/aes_cipher_top_20240910_1612_area.rpt
report_qor > ../report/aes_cipher_top_20240910_1612_qor.rpt
report_timing -max_paths 10000 > ../report/aes_cipher_top_20240910_1612_report_timing.rpt
report_constraint -all_violators -nosplit > ../report/aes_cipher_top_20240910_1612_report_constraint.rpt
report_power > ../report/aes_cipher_top_20240910_1612_power.rpt

```

### 3. 运行设计

运行设计的方式为 source run.sh “X” 运行如下图所示

[eda@edacentos DC\_FLOW]\$ source run.sh "AES\_100M"，设置运行 DC 的 log 名称为 AES\_100M.log。

### 4. 运行结果

运行后可以看到在 work 文件夹下有对应名字的 log 文件，保存了逻辑综合的信息

```

[eda@edacentos work]$ ls *log
AES_100M_20240910_1453.log  command.log

```

对应的在 DC\_FLOW 文件夹下，创建了 output 和 report 两个文件夹，分别保存了逻辑综合后的网表文件，sdc 文件，以及时序，面积等信息的报告，如下图所示

```
[eda@edacentos SYN_FLOW]$ cd report/
[eda@edacentos report]$ ls
aes_cipher_top_20240910_1453_area.rpt
aes_cipher_top_20240910_1453_power.rpt
aes_cipher_top_20240910_1453_qor.rpt
aes_cipher_top_20240910_1453_report_constraint.rpt
aes_cipher_top_20240910_1453_report_timing.rpt
[eda@edacentos output]$ ls
aes_cipher_top_20240910_1628_compile.ddc  aes_cipher_top_20240910_1628.sdf
aes_cipher_top_20240910_1628_link.ddc    aes_cipher_top_20240910_1628.svf
aes_cipher_top_20240910_1628.sdc         aes_cipher_top_20240910_1628.v
```