

# آشنایی با ابزار سنتز Design Compiler و نحوه سنتز مدار توسط آن

## آشنایی با ابزار Power Compiler

تهیه و تنظیم: صبا احمدیان ahmadian@ce.sharif.edu

### ابزار های مورد استفاده:<sup>۱</sup>

Synopsys Design Compiler  
Synopsys Power Compiler  
ModelSIM

### مقدمه (مهم):

ابزار Design Compiler ابزاری ست که برای سنتز طراحی های ما در زبان های Verilog, VHDL مورد استفاده قرار میگیرد. طراحی که ما به زبان ورایلاگ پیاده سازی میکنیم، توسط این ابزار به سلولهای استاندارد کتابخانهی استاندارد سنتز میشود. اگر عملیات سنتز به درستی صورت گیرد، تغییری در عملکرد آن بوجود نخواهد آمد. فقط طراحی ما شامل خواهد شد از یک سری instance از سلولهای کتابخانه مورد نظر. طراحی که به درستی سنتز شود یا بعبارتی سنتز پذیر باشد را در واقعیت با استفاده از سلولهای استاندارد، می توان ساخت. بسته به طراحی ما ابزار سنتز می تواند طرح بزرگ و پیچیده ی ما را ساده و بهینه کند، یا اینکه طرح ساده ما را پیچیده تر کند.

طراحی را می توان قبل و یا بعد از سنتز تست یا شبیه سازی یا Simulate کرد که توسط ابزارهای شبیه سازی نظیر modelsim و ... صورت می پذیرد. در هنگام شبیه سازی با دادن یک سری ورودی به طرح از کارکرد صحیح آن مطلع می شویم. (سنتز و شبیه سازی دو موضوع جدا از هم هستند).

پس از آنکه طرحمان سنتز شد و از صحت کارکرد آن پس از سنتز اطمینان حاصل کردیم، میتوانیم اطلاعات دقیق تری نظیر تاخیر مسیر بحرانی، مساحت، توان و سایر اطلاعات مفید را بدست آورد. حتی می توان در حین سنتز محدودیتهایی اعمال کرد که طرح سنتز شده دارای ویژگی های خاصی نظیر حداقل تاخیر، یا حداقل مساحت یا توان را داشته باشد.

در بخش های آینده همگی این موارد به تفصیل شرح داده خواهد شد.

### موارد مورد نیاز برای سنتز:

- ۱- کد Verilog , VHDL طرح مورد نظر
- ۲- کتابخانه با فرمت db.
- ۳- میتوان تمامی دستورات زیر را در فایل script با فرمت scr نوشت و همه آنها را با انتخاب گزینه execute script یکجا اجرا کرد.

<sup>۱</sup> میتوان از ماشین های مجازی ICIC یا Ubuntu ی آقای احمدیان یا Ubuntu ی آقای محمدی استفاده کرد. icic جدیدتر از موارد دیگر است. هم چنین مورد آخر در cabinet هست.

## نکات مهم و کاربردی:

- ۱- برای بالا آمدن DC به صورت گرافیکی دستور `design_vision` را در ترمینال وارد میکنیم.
- ۲- برای تبدیل کتابخانه با فرمت `.lib` به `.db`. از دستورات زیر در ترمینال DC استفاده می‌کنیم:

```
read_lib  
write_lib
```

## شرح دستورات ورودی برای سنتز:

**توجه -** تمامی دستورات مورد نیاز برای سنتز به صورت یک جا در انتهای فایل آورده شده است، که میتوان به عنوان یک فایل با فرمت `.sct`. از آن استفاده کرد.

**توجه مهم تر -** این بخش به توضیح دستورها می‌پردازد، و تقریباً قابل حذف کردن هست. در صورت حذف این بخش، تنها مواردی که در انتهای فایل آورده شده را رعایت کرده و موارد گفته شده را عوض می‌کنیم. (صفحه آخر)

در ابتدا بایستی فایل یا فایل‌هایی که میخواهیم سنتز کنیم را اعلام کنیم:

```
set my_files <Ur_verilog_file_name>.v
```

اگر تعداد آنها بیش از یکی است: (مثلا اگر یک ALU داریم که دارای زیر بخشهای `add`, `sub`, `mult`,... در فایل‌های جداگانه است، نیاز است که نام همه فایلها آورده شود؛ اما اگر تمامی ماژولها داخل یک فایل است، فقط اسم همان فایل را می‌آوریم).

```
set my_files [list <Ur_verilog_file_name_1>.v <Ur_verilog_file_name_2>.v]
```

**توجه -** فایل `testbench` را اینجا وارد نمی‌کنیم!! برای سنتز به آن نیاز نیست. اگر هم وارد کنیم DC آنرا نادیده می‌گیرد. همانطور که از نام آن پیداست به عنوان یک طراحی در سطح یا طبقه‌ی بالاتر است برای تست طراحی اصلی ما، هدف ما این است که طرح خودمان را سنتز کنیم سپس برای کسب اطلاع از درستی آن، آن را تست یا `Simulate` کنیم که طرح سنتز شده بعداً در ابزار شبیه سازی نظیر `modelsim` مورد تست قرار می‌گیرد.

سپس برخی **متغیرها** نظیر `top level module` و کلاک و فرکانس آن را مشخص می‌کنیم، در صورتی که ماژولمان ورودی کلاک داشته باشد بایستی فرکانس کلاک ماژول کمتر یا مساوی فرکانس سنتز باشد.

```
set my_toplevel <top_module_name>
```

**توجه -** اینجا اسم بالاترین ماژول را وارد می‌کنیم. نه اسم فایل وریلاگ (البته اگر متفاوت باشند!). یعنی اگر اسم فایلمان `ALU.v` باشد ولی اسم ماژول باشد: `myALU`، خواهیم نوشت: `set my_toplevel myALU`

```
set my_clock_pin clk
```

```
set my_clk_freq_MHz 100
```

همچنین در صورت نیاز می‌توانیم برای ورودی‌ها و خروجی‌ها تأخیر قرار دهیم.

```
set my_input_delay_ns 0
```

```
set my_output_delay_ns 0
```

سپس کتابخانه‌هایی که قرار است طرحمان از سلولهای آن استفاده کند و به آن سنتز شود را تعیین می‌کنیم. (همان کتابخانه‌های استاندارد که در مقدمه بحث شد).

```
set link_library "<db_lib_name>.db"
```

```
set target_library "<db_lib_name>.db"
```

سپس برای کامپایل اولیه و رفع خطاهای احتمالی از دستور `analyze` به ترتیب زیر استفاده می‌کنیم:

```
analyze -f Verilog $my_files #(change it to vhdl if your design is in vhdl format)
```

```
elaborate $my_toplevel
```

```
current_design $my_toplevel
```

```
link
```

سپس بایستی کلاک مدار و سرعت آنرا مشخص کنیم:

```
set my_period [expr 1000 / $my_clk_freq_MHz]
```

```
set find_clock [find port [list $my_clock_pin]]
```

```
if{ $find_clock != [list] } {
```

```
    set clk_name $my_clock_pin
```

```
    create_clock -period $my_period $clk_name
```

```
} else {
```

```
    set clk_name vclk
```

```
    create_clock -period $my_period -name $clk_name
```

```
}
```

مشخص کردن میزان `delay` ورودی و خروجی:

```
set_input_delay $my_input_delay_ns -clock $clk_name [remove_from_collection [all_inputs]  
$my_clock_pin]
```

```
set_output_delay $my_output_delay_ns -clock $clk_name [all_outputs]
```

## در نهایت می‌رسیم به مرحله سنتز:

برای ایجاد محدودیت‌هایی نظیر مساحت و توان و سرعت (راجع به سرعت بعداً صحبت خواهد شد) دستورات مورد نیاز را در این مرحله وارد میکنیم:

### ایجاد محدودیت مساحت:

```
#area
```

```
set_max_area 0
```

با این دستور است که ابزار سنتز سعی میکند تا تعداد گیتها را بهینه کند.

### ایجاد محدودیت توان:

```
#power
```

```
set_max_dynamic_power <value, you can set it to zero>
```

```
set_max_static_power <value, you can set it to zero>
```

```
compile
```

```
check_design
```

**نکته -** دستور کامپایل option های فراوانی دارد، با جستجو کردن عبارت design compiler command در جستجوگر گوگل و دانلود اولین فایل pdf پیشنهاد شده و جستجوی کلمه‌ی compile داخل این فایل، این option ها با توضیحات دقیق در آنجا قرار دارند، با توجه به نیازتان می‌توانید از آنها استفاده کنید یا نکنید (همین کامپایل خالی هم کافیه).

در اینجا طرح سنتز شده است و میتوان گزارشهای مورد نیاز را از آن گرفت. (دستورات پر مصرف در پایین آمده است)

### ایجاد محدودیت تاخیر یا سرعت:

با استفاده از دستور report\_timing میتوان تمامی اطلاعات مربوط به ماکزیمم تأخیر و ... را بدست آورد. پس از وارد کردن این دستور در گزارش مربوطه، start point و end point برای CP (critical path) قرار دارد؛ پس از بدست آوردن ابتدا و انتهای CP دستور زیر را وارد میکنیم تا محدودیت زمانی (بعبارتی سرعت) را ایجاد کنیم (البته باید در نظر داشته باشیم که برای پیدا کردن مسیر بحرانی باید محدودیت دیگری نظیر مساحت و توان و ... نداشته باشیم):

```
set_max_delay 5 -from start_point -to end_point
```

```
 #(5 is an example, you can change it, unit is "ns", be aware of violation!)
```

توجه - در گذاشتن این محدودیت باید به این نکته توجه داشت که مقدار به طوری تعیین شود که مدار بتواند در آن کلاک پاسخگو باشد. در همین راستا، پس از ایجاد محدودیت و اتمام سنتز، دوباره گزارش زمانی میگیریم و آنرا نگاه میکنیم و slack time را دقت میکنیم که منفی نشود. و به عبارتی مدار بتواند پاسخگو باشد.

## گرفتن گزارش:

دستورات زیر نیز یک سری دستورات پرکاربرد برای گرفتن گزارش میباشند:

نکته - برای داشتن گزارشها را در فایل، میتوان چنین عمل کرد:  
report\_timing > ./your path/timing\_report.txt  
نکتهی مهمتر - بعد از دستور report\_timing یک space قرار میگیرد، بعد > بعدش دوباره space بعد بقیهش.

report\_timing

report\_area

report\_qor #qor: Quality Of Result

report\_qor تقریباً حاوی آنچه که ما میخواهیم هست! (مسیر بحرانی و مساحت و عمق مدار و ...)

مهم - همچنین میتوان (البته باید!) نت لیست تولید شده حاصل از سنتز را نیز با فرمتهای vhd , verilog دریافت کرد (نت لیست عبارتست از یک کد که همان عملکرد کد قبلی خودمان را دارد با این تفاوت که به cellهای استاندارد کتابخانه مورد نظر نگاشته شده است.) و می توان در مرحلهی post-synthesis-simulation از آنها بهره برد.

## مرحله post synthesis simulation:

در این مرحله نت لیست حاصل از سنتز را به همراه کتابخانه با فرمت v. یا vhd. و یک ماژول تست، در modelsim شبیه سازی می کنیم تا از صحت کارکرد آن پس از سنتز اطمینان پیدا کنیم و بدانیم که آیا سنتز به درستی انجام شده یا نه؟!

نکته - در صورت مواجهه با هرگونه error های عجیب و غریب در حین استفاده از فرمت v. در این مرحله، میتوانیم به راحتی از نت لیست vhdl + فایل vhdl کتابخانه استفاده کرد و ماژول تست همان وریلاگ بماند.

## محاسبه توان با استفاده از Power Compiler:

برای محاسبه توان روشهای مختلفی وجود دارد، اینجا به شرح یکی از آنها که پرکاربردتر و دقیقتر است؛ می پردازیم.

این مرحله پس از سنتز صورت میگیرد و **نت لیست صحیح حاصل** از سنتز **پیش نیاز** آن می باشد. با این تفاوت که نیازمند فایلی با فرمت vcd (vcd = **V**alue **C**hange **D**ump) هستیم تا میزان activity طرح را به power compiler وارد کنیم. activity نشان میدهد که هر یک از گره ها یا خازنهای مدار چقدر صفر و یک شده اند. (همان آلفا ست در فرمول توان پویا). برای این کار نیاز است تا طرحمان به ازای انواع ورودیها کار کند و خروجی تولید کند. و تمامی گره های آن به کار گرفته شود. پس نیاز داریم آن را در ابزار شبیه سازی نظیر modelsim شبیه سازی کنیم با انواع ورودی ها، تا به activity آن پی ببریم.

بنابراین همان کارهای مرحله post synth sim را انجام میدهیم به این ترتیب که پس از وارد کردن دستور vsim -novopt (همان simulate without optimization) پس از اینکه شبیه سازی انجام شد (در طرحهای بزرگ ممکن است طولانی شود) دستورهای زیر را به ترتیب وارد میکنیم:

vcd file file.vcd

این دستور یک فایل با اسم file با فرمت vcd خالی ایجاد می کند. محتویات فایل vcd به این ترتیب است که برای مثال مقدار سیگنال a در t1 برابر با یک و در زمان t2 برابر با صفر است و ... برای اضافه شدن سیگنالها به فایل vcd دستور زیر را وارد میکنیم:

vcd add -r /testbench\_name/instance\_name/\*

**نکته -** اگر فایلی داریم که اسمش top.v است و درون آن ماژولی است که اسم ماژول JJ است و درون این ماژول از طراحی اصلی (که مثلا اسمش myALU است، که سه ساعته داریم سنتزش میکنیم) instance گرفته شده است. و اسم instance برای نمونه myINST است. به ترتیب بجای testbench\_name می نویسیم: top و به جای instance\_name می نویسیم: myINST

```
module jj;  
....  
myALU myINST (.....);  
....  
endmodule
```

این چند سطر درون فایلی با نام top.v در کامپیوتر شما به صورت ذخیره شده قرار دارند.

سپس دستور run 100ns را وارد میکنیم. (پیش از وارد کردن دستور به توجه زیر توجه کنید). تا شبیه سازی کامل شده و تغییرات مقادیر سیگنالها به فایل vcd وارد شود.

**توجه -** مدت زمانیکه در دستور run وارد میشود با توجه به طرح و پریود کلاک و تعداد تست هایی که میخواهیم به طرح بدهیم میتواند متغیر باشد، برای آنکه توان گزارش شده معقول تر باشد این مدت زمان را زیاد میدهیم، عبارتی تستهای بیشتر و متنوعتری به مدار میدهیم.

اکنون با دستور quit-sim از محیط شبیه سازی خارج میشویم؛ فایل vcd بدست آمده را با دستور vcd2saif به فرمت saif تبدیل میکنیم. (saif = **S**witching **A**ctivity **I**nterchange **F**ormat)

فایل saif که از روی vcd بدست آمده است حاوی اطلاعاتی نظیر: مدت زمانی که اندازه سیگنال صفر، یک، X بوده و... و همچنین تعداد transitionها به صفر و یک و ... می باشد. (نیاز اصلی به همین تعداد transition هاست). این دستور را در ترمینال معمولی میزنیم:

```
vcd2saif -i file.vcd -o file.saif
```

**نکته -** حجم فایل saif در مقایسه با vcd بسیار کم و درک و استخراج اطلاعات از آن آسان تر می باشد.

اکنون فایل saif بدست آمده را به power compiler میدهیم تا توان محاسبه شود.

### بیشتر بدانیم!

**مهم -** توجه شود که برای محاسبه توان، طرح سنتز شده باید روی Design Compiler باشد. لذا پس از اتمام عملیات سنتز از ترمینال مربوطه خارج نمی شویم. اگر هم اشتباهاً خارج شدیم میتوان یا دوباره سنتز کرد یا از فایل db بهره برد.

```
read_db mydesign.db
```

برای تولید db. کفایست در حین سنتز و هنگام گرفتن report دستور نوشتن طرح روی فایل db را به ترتیب زیر وارد کرد:

```
write -f db -o mydesign.db
```

**مهم تر -** در ورژنهای جدید DC این دستور نوشتن support نمیشود. ولی یک فایل دیگر می توان بدست آورد با فرمت ddc. که متأسفانه به طور کامل کار db. را نمیکند، و کلاً به درد نمیخورد! ما سعی می کنیم راه ساده تر را انتخاب کنیم و از DC و ترمینالش خارج نشویم. البته در صورت علاقه برای تولید و خواندن ddc. نیز چنین می کنیم:

```
write -f dcc -o mydcc.dcc
```

```
read_dcc mydcc.dcc
```

با دستور زیر میتوان فایل saif را به power compiler داد:

```
read_saif -input file.saif -instance testbench_name/instance_name
```

**نکته -** حتما رجوع شود به نکته ی موجود در وسط صفحه ی ۶.

با return شدن مقدار یک متوجه میشویم که عملیات read به درستی صورت گرفته است. (یک ۱ می نویسد).

سپس با وارد کردن دستور report\_power میتوان میزان توان dynamic , static leakage و ... را بدست آورد.

```
report_power > ./your path/ power_report.txt
```

## پیوست:

همین را copy و سپس paste کنید، قسمتهای قرمز را عوض کنید، با فرمت scr. ذخیره کنید. سپس در محیط گرافیکی DC که با وارد کردن دستور design\_vision در ترمینال بالا می آید. به قسمت file در گوشه سمت چپ – بالا رفته و در منوی پایین آمده، execute script را انتخاب کرده و اجرا کنید. به محل قرار گرفتن کتابخانه ها و کدها و ... توجه داشته باشید.

**نکته –** این فایل script برای یک سنتز کاملاً معمولی است و هیچ دستور خاصی ندارد، دستورات مورد نیاز را یا از توضیحات قبلی بدست آورده و در اینجا اضافه کنید یا خود جستجو کنید.

```
set my_files [list xxx.v yyy.v]
set my_toplevel zzz
set my_clock_pin CLK
set my_clk_freq_MHz 100
set my_input_delay_ns 0
set my_output_delay_ns 0
set link_library /path/ ttt.db
set target_library /path/ ttt.db
define_design_lib WORK -path ./WORK
analyze -f verilog $my_files
elaborate $my_toplevel
current_design $my_toplevel
link
uniquify
set my_period [expr 1000 / $my_clk_freq_MHz]
set find_clock [ find port [list $my_clock_pin] ]
if { $find_clock != [list] } {
    set clk_name $my_clock_pin
    create_clock -period $my_period $clk_name
} else {
    set clk_name vclk
    create_clock -period $my_period -name $clk_name
}
set_input_delay $my_input_delay_ns [all_inputs]
set_output_delay $my_output_delay_ns [all_outputs]
compile
check_design
report_timing > /path/time_report.txt
report_area > /path/area.txt
report_qor > /path/qor.txt
write -f db -output /path/mydb.db
write -f verilog -output / path /netlist.v
write -f vhdl -output / path /netlist.vhd
```