

# Integrated Circuit Design

## Homework #4 Logic Synthesis with Design Compiler

Due: 2018/05/23 13:00

### 1. 問題描述

接續作業三之問題，利用 Design Compiler 將上次作業 Register Transfer Level 的程式碼轉成 Gate Level 的檔案。

### 2. Design Compiler 簡介

在積體電路設計中，邏輯合成(logic synthesis)是將 RTL 之數位電路描述，經過布林函數化簡、優化後，轉換到 Gate Level 電路 Netlist 的過程。

### 3. 相關檔案介紹

#### (1) .synopsys\_dc.setup

開啟 Design Compiler 時，會自動讀取同個路徑底下的隱藏檔.synopsys\_dc.setup，該檔案內部定義了執行 Design Compiler 時所需 library 的路徑與名稱，執行 Design Compiler 前請用 ls -a 指令確認此檔案有在當前目錄底下。

下圖為檔案內最核心的部分，這些 library 包含對應合成前後的元件、時間資訊(Setup time...)與元件的功能(邏輯上的 AND, OR, NOT...等運算)。

```
set search_path      "/cad/cell_library/CBDK_TSMC90G_Arm/CIC/SynopsysDC/db $search_path"
set target_library   "slow.db fast.db"
set link_library     "* $target_library dw_foundation.sldb"
set symbol_library   "generic.sdb"
set synthetic_library "dw_foundation.sldb"
```

#### (2) tsmc13\_neg.v

這個檔案是 TSMC 130 奈米製程的 cell library，裡面描述了各種標準元件(standard cell)的電路特性，當完成電路的邏輯合成後，須將此檔案與合成後的 Gate level 檔案一起進行 post-synthesis 的模擬。

```
module BUF3 (Y, A);
output Y;
input A;

  buf I0(Y, A);
  specify
    // delay parameters
    specparam
      tplh$A$Y = 1.0,
      tphl$A$Y = 1.0;

    // path delays
    (A *> Y) = (tplh$A$Y, tphl$A$Y);
  endspecify
endmodule // BUF3
```

### (3) IG.sdc

Sdc 檔(standard design constraints file)內規範了 Design Compiler 合成電路時必須遵守的時間資訊，也就是 Design Compiler 在做電路優化時必須達到的目標。Constraints 可以簡單分為兩類，分別是 Optimization Constraints 以及 Design Rule Constraints。

#### A. Optimization Constraints

```
set cycle 5.0
set t_in [expr $cycle/2]
set t_out 0

# clock constraints
create_clock -period $cycle [get_ports clk]
set_dont_touch_network [get_clocks clk]
set_fix_hold [get_clocks clk]
set_ideal_network [get_ports clk]
set_clock_uncertainty 0.1 [get_clocks clk]
set_clock_latency 0.5 [get_clocks clk]
set_input_transition 0.5 [all_inputs]
set_clock_transition 0.1 [all_clocks]
set_input_delay $t_in -clock clk [remove_from_collection [all_inputs] [get_ports clk]]
set_output_delay $t_out -clock clk [all_outputs]
```

- (A) create\_clock: 表示合成出之電路的時脈週期，單位為 ns。
- (B) set\_dont\_touch\_network [get\_clocks clk]: 表示在進行邏輯合成時，暫時不考慮 clock 線路上的 fanout 是否過載，也不對 clock 線路插入推動 buffer。
- (C) set\_fix\_hold: 修正 hold time violation
- (D) set\_ideal\_network: 暫時將 clock 線路設定為理想電路(無負載、無延遲)
- (E) set\_clock\_uncertainty: 設定 clock skew
- (F) set\_clock\_latency: 線路造成之 clock delay
- (G) set\_input\_transition, set\_clock\_transition: 電壓從 0 到 1 以及從 1 到 0 所需時間(精確來說是超過 50% 線所需充放電時間)
- (H) set\_input\_delay, set\_output\_delay: 整個電路的輸入輸出端需保留的延遲時間，讓前一級或後一級的電路留有運算時間。

#### B. Design Rule Constraints

```
# 6 environment settings
set_operating_conditions -min fast -max slow
set_load 4 [all_outputs]
set_wire_load_model -name tsmc13_wl10 -library slow
```

- (A) set\_operating\_condition: 在使用晶片時，需考慮 process, voltage, temperature(PVT)的變化。
- (B) set\_load: 電路輸出端負載不會是 0，因此要預先設定。
- (C) set\_wire\_load\_model: 設定連接各個元件的線所需面積以及時間。

#### 4. 操作流程

- (1) 將 hw4.zip 解壓縮到 hw4 資料夾
- (2) 將 hw3 的 IG.v、testfixture.v、dat 資料夾複製到 hw4 資料夾底下
- (3) 將 synopsys\_dc.setup 檔重新命名為 .synopsys\_dc.setup

※若有需要可以自行修改檔案內的 library 路徑

※注意檔名第一個字元必須有「.」

- mv synopsys\_dc.setup .synopsys\_dc.setup
- (4) source /usr/cad/synopsys/CIC/synthesis.cshrc
  - (5) 開啟 design compiler
    - GUI 版：dv &
    - 無 GUI 版：dc\_shell
  - (6) 讀取作業三所寫之 RTL 電路
    - read\_file -format verilog "./IG.v"

注意：合成結果不能有 latch，若有 latch 須重新修改 Hw3 的 RTL code

Register Name	Type	Width	Bus	MB	AR	AS	SR	SS	ST
y_jump_flag_reg	Flip-flop	8	Y	N	N	N	N	N	N
read_state_reg	Flip-flop	2	Y	N	N	N	N	N	N
grad_do_r_reg	Flip-flop	20	Y	N	N	N	N	N	N
done_r_reg	Flip-flop	1	N	N	N	N	N	N	N
grad_addr_r_reg	Flip-flop	16	Y	N	N	N	N	N	N
write_buf_reg	Flip-flop	10	Y	N	N	N	N	N	N
write_buf_reg	Flip-flop	10	Y	N	N	N	N	N	N
grad_wr_r_reg	Flip-flop	1	N	N	N	N	N	N	N
img_rd_r_reg	Flip-flop	1	N	N	N	N	N	N	N
img_addr_r_reg	Flip-flop	16	Y	N	N	N	N	N	N
read_buf_reg	Flip-flop	8	Y	N	N	N	N	N	N
read_buf_reg	Flip-flop	8	Y	N	N	N	N	N	N
read_buf_reg	Flip-flop	8	Y	N	N	N	N	N	N
state_reg	Flip-flop	2	Y	N	Y	N	N	N	N
state_reg	Flip-flop	1	N	N	N	Y	N	N	N
x_jump_flag_reg	Flip-flop	8	Y	N	N	N	N	N	N
addr_count_reg	Flip-flop	16	Y	N	N	N	N	N	N

上圖中，type 需皆為 Flip-flop，請在報告中附上截圖。

- (7) 讀取 standard design constraints file
  - source -echo -verbose ./IG.sdc
- (8) 執行以下三行指令以修正合成時可能遇到的問題
  - set high\_fanout\_net\_threshold 0
  - uniquify
  - set\_fix\_multiple\_port\_nets -all -buffer\_constants [get\_designs \*]
- (9) 開始合成 gate level 電路
  - compile
- (10) 檢查 critical path 所需時間是否小於 cycle time(slack $\geq$ 0)，若 critical path 大於 cycle time(slack $<$ 0)，則代表 design compiler 無法將你的電路用這個 cycle time 合成，請去修改 IG.sdc 中「set cycle」後的數字或是修改 RTL code，並重新執行步驟(4)~(9)。
  - report\_timing

- (11) 執行以下指令以修正輸出檔案時可能發生之錯誤
- `remove_unconnected_ports -blast_buses [get_cells -hierarchical *]`
  - `set bus_inference_style {%s[%d]}`
  - `set bus_naming_style {%s[%d]}`
  - `set hdlout_internal_busses true`
  - `change_names -hierarchy -rule verilog`
  - `define_name_rules name_rule -allowed {a-z A-Z 0-9 _} -max_length 255 -type cell`
  - `define_name_rules name_rule -allowed {a-z A-Z 0-9 _[]} -max_length 255 -type net`
  - `define_name_rules name_rule -map {{\\"*cell\\"} "cell"}}`
  - `define_name_rules name_rule -case_insensitive`
  - `change_names -hierarchy -rules name_rule`
- (12) `report_area`
- (13) `report_power`
- (14) 輸出 gate level 之 verilog code
- `write -format verilog -hierarchy -output "IG_syn.v"`
- (15) 輸出電路延遲資訊檔(standard delay format)
- `write_sdf -version 1.0 -context verilog -load_delay net IG_syn.sdf`
- (16) quit, 離開 design compiler
- (17) 修改 testfixture.v 中第二行`define SDFFILE 後的路徑, 改成"IG\_syn.sdf"。
- (18) 修改 testfixture.v 中第三行`define cycle 後的數字, 改成 sdc file 中設定的 cycle time, **sdc 中的 cycle time 必須和 testbench 一樣才是有效的模擬。**
- (19) 用 ncverilog 對 design compiler 輸出之 SET\_syn.v 進行 gate level 模擬
- Gate-level 模擬指令如下:
- `ncverilog testfixture.v -v tsmc13_neg.v IG_syn.v +access+r +define+SDF`

## 5. 檔案說明

檔名	說明
IG.sdc	除了 cycle time 以外, 其他參數請勿修改。
synopsys_dc.setup	Design compiler 環境設定, 上傳至工作站後請記得修改檔名為 .synopsys_dc.setup
tsmc13_neg.v	模擬用之 Cell library

表三、檔案說明

## 6. 模擬結果

如果模擬有錯誤，testbench 會顯示期望的結果跟實際的結果:

```
Gradient y of pixel      0 is wrong! Your gradient y is 001, but expected gradient y is 3ff
Gradient x of pixel      1 is wrong! Your gradient x is 000, but expected gradient x is 3ff
Gradient y of pixel      1 is wrong! Your gradient y is 000, but expected gradient y is 001
Gradient x of pixel      2 is wrong! Your gradient x is 002, but expected gradient x is 000
Gradient x of pixel      3 is wrong! Your gradient x is 3fe, but expected gradient x is 002
Gradient x of pixel      4 is wrong! Your gradient x is 3fd, but expected gradient x is 3fe
Gradient x of pixel      5 is wrong! Your gradient x is 001, but expected gradient x is 3fd
Gradient y of pixel      5 is wrong! Your gradient y is 001, but expected gradient y is 000
Gradient x of pixel      6 is wrong! Your gradient x is 3fd, but expected gradient x is 001
Gradient y of pixel      6 is wrong! Your gradient y is 000, but expected gradient y is 001
Gradient x of pixel      7 is wrong! Your gradient x is 004, but expected gradient x is 3fd
```

```
Gradient of Pixel: 0 ~    999 are wrong ! The wrong pixel reached a total of    1594 or more !
Gradient of Pixel: 0 ~   1999 are wrong ! The wrong pixel reached a total of   3294 or more !
Gradient of Pixel: 0 ~   2999 are wrong ! The wrong pixel reached a total of   5016 or more !
Gradient of Pixel: 0 ~   3999 are wrong ! The wrong pixel reached a total of   6748 or more !
Gradient of Pixel: 0 ~   4999 are wrong ! The wrong pixel reached a total of   8451 or more !
Gradient of Pixel: 0 ~   5999 are wrong ! The wrong pixel reached a total of  10192 or more !
```

```
-----
FAIL! There are      117677 errors at functional simulation !
----- The test result is ..... FAIL -----
```

如果模擬無法在規定 cycle 數完成，會顯示以下結果:

```
-----
Error!!! Running out of time!

There is something wrong with your code!
-----The test result is .....FAIL -----
-----
Simulation complete via $finish(1) at time 100 MS + 0
```

如果模擬結果正確，testbench 會顯示以下結果:

```
=====The test result is ..... PASS=====

*****
**                                     **
**          Congratulations !!          **
**                                     **
** All data have been generated successfully! **
**                                     **
*****

                                     0,0
                                     ^ ^ ^
                                     | ^ ^ |
                                     \ m m /
```

## 7. 作業評分

本次作業滿分標準為：

- (1) 合成結果不能有 latch
- (2) 通過 Gate-level 模擬，並同時符合以下條件
  - 面積小於 20000  $\mu\text{m}^2$
  - cycle time 小於 10 ns

若你 Hw3 的 RTL code 無法修改到符合上述條件，可以寄信給助教，助教會給你另外一份可以合成的 RTL 檔案，完成電路合成後，可以拿到 60 分。

Report 內需要包含下列項目：

- (1) 證明電路中沒有 Latch 的截圖
- (2) report\_timing 的截圖
- (3) report\_area 的截圖
- (4) report\_power 的截圖
- (5) Gate level 模擬通過截圖

所需繳交的檔案為：

Type	File Name	Description
RTL-level	IG.v	合成所使用的 RTL Verilog Code
Gate-level	IG _syn.v	Gate-level netlist
	IG _syn.sdf	Standard Delay format file
Report	ICD_HW4_StudentID.pdf	內需包含 Gate level 模擬結果的截圖、合成沒有 latch 的截圖、design complier 所生成的 timing/area/power report

請將以上檔案壓縮成 YourID#\_HW4\_v\*.tar.gz 並使用 FTP 繳交

[FTP info]

IP: 140.112.48.159      port: 56666

User name: icd2018      Password: ntueeicd

[助教範例]

(1) report\_timing

Des/Clust/Port	Wire Load Model	Library	
IG	tsmc13_wl10	slow	
Point	Incr	Path	
clock clk (rise edge)	0.00	0.00	
clock network delay (ideal)	0.50	0.50	
img_addr_reg_1_/CK (DFFRX2)	0.00	0.50 r	
img_addr_reg_1_/Q (DFFRX2)	0.43	0.93 f	
sub_70/A[1] (IG_DW01_dec_0)	0.00	0.93 f	
sub_70/U10/Y (OR2X2)	0.22	1.15 f	
sub_70/U13/Y (OR2X4)	0.15	1.30 f	
sub_70/U24/Y (OR2X1)	0.25	1.54 f	
sub_70/U3/Y (OR2X1)	0.28	1.82 f	
sub_70/U16/Y (OR2XL)	0.32	2.14 f	
sub_70/U4/Y (OR2XL)	0.34	2.48 f	
sub_70/U5/Y (OR2XL)	0.34	2.82 f	
sub_70/U2/Y (OR2X1)	0.31	3.13 f	
sub_70/U19/Y (NOR2X1)	0.34	3.47 r	
sub_70/U15/Y (NAND2X1)	0.22	3.69 f	
sub_70/U8/Y (OR2XL)	0.34	4.03 f	
sub_70/U1/Y (NOR2X1)	0.36	4.40 r	
sub_70/U14/Y (NAND2X1)	0.23	4.63 f	
sub_70/U26/Y (NOR2BX1)	0.17	4.80 r	
sub_70/U17/Y (XOR2XL)	0.27	5.07 r	
sub_70/SUM[15] (IG_DW01_dec_0)	0.00	5.07 r	
U296/Y (A022XL)	0.23	5.29 r	
grad_addr_reg_15_/D (DFFRX2)	0.00	5.29 r	
data arrival time	clock time		5.29
clock clk (rise edge)	5.00	5.00	
clock network delay (ideal)	0.50	5.50	
clock uncertainty	-0.10	5.40	
grad_addr_reg_15_/CK (DFFRX2)	0.00	5.40 r	
library setup time	-0.10	5.30	
data required time		5.30	
data arrival time	Slack 必須 >=0		-5.29
slack (MET)		0.00	



## (2) report\_area

```
*****
Report : area
Design : IG
Version: G-2012.06
Date   : Tue May 1 03:15:03 2018
*****

Library(s) Used:

    typical (File: /home/raid7_2/course/cvsd/CBDK_IC_Contest/CIC/SynopsysDC/db/typical.db)

Number of ports:          95
Number of nets:          398
Number of cells:         323
Number of combinational cells: 254
Number of sequential cells:  67
Number of macros:         0
Number of buf/inv:       115
Number of references:     33

Combinational area:      4201.064915  combinational 電路面積
Noncombinational area:  2298.279560  Sequential (register)電路面積
Net Interconnect area:  43633.552185

Total cell area:        6499.344475  Gate level 電路面積
Total area:             50132.896660
```

## (3) report\_power

```
Design      Wire Load Model      Library
-----
IG          tsmc13_wl10          slow

Global Operating Voltage = 1.08
Power-specific unit information :
  Voltage Units = 1V
  Capacitance Units = 1.000000pf
  Time Units = 1ns
  Dynamic Power Units = 1mW      (derived from V,C,T units)
  Leakage Power Units = 1pW

  Cell Internal Power   = 443.2127 uW   (20%)
  Net Switching Power   = 1.7979 mW   (80%)
  -----
  Total Dynamic Power   = 2.2411 mW   (100%)
  Cell Leakage Power    = 1.3611 uW

Power Group      Internal      Switching      Leakage      Total
                  Power          Power          Power          Power  ( % ) Attrs
-----
io_pad           0.0000          0.0000          0.0000          0.0000  ( 0.00%)
memory           0.0000          0.0000          0.0000          0.0000  ( 0.00%)
black_box        0.0000          0.0000          0.0000          0.0000  ( 0.00%)
clock_network    0.0000          0.0000          0.0000          0.0000  ( 0.00%)
register          0.3906          4.4794e-02      5.7142e+05      0.4359  ( 19.44%)
sequential       0.0000          0.0000          0.0000          0.0000  ( 0.00%)
combinational    5.2633e-02      1.7531          7.8967e+05      1.8065  ( 80.56%)
-----
Total            0.4432 mW       1.7979 mW       1.3611e+06 pW   2.2425 mW
```



(4) Gate level 模擬通過

```
=====The test result is ..... PASS=====

*****
**                                     **
**          Congratulations !!          **
**                                     **
** All data have been generated successfully! **
**                                     **
*****

          /|_/_|
         / 0,0 \
        / ^ ^ ^ \
       / ^ ^ ^ ^ \w
      / m   m   \

=====
Simulation complete via $finish(1) at time 655398868 PS + 0
./testfixture.v:147      $finish;
ncsim> exit
```

模擬總時間

電路 cycle time 為 5ns，總模擬時間約為 655400 ns，完成電路運算共需 131080 cycles