

HDL Homework2 說明

邱奕綸 chiu91523@gmail.com

2025 HDL

HW2資料夾建立規則

- 先在HW2底下 建立個別的資料夾

Name	Size (KB)	Last modified	Owner	Group
..				
add_sub		2025-09-26...	m133040041	m133040041
clock_gating		2025-09-25...	m133040041	m133040041
compare		2025-09-25...	m133040041	m133040041
non_pipeline		2025-09-25...	m133040041	m133040041
pipelined		2025-09-25...	m133040041	m133040041

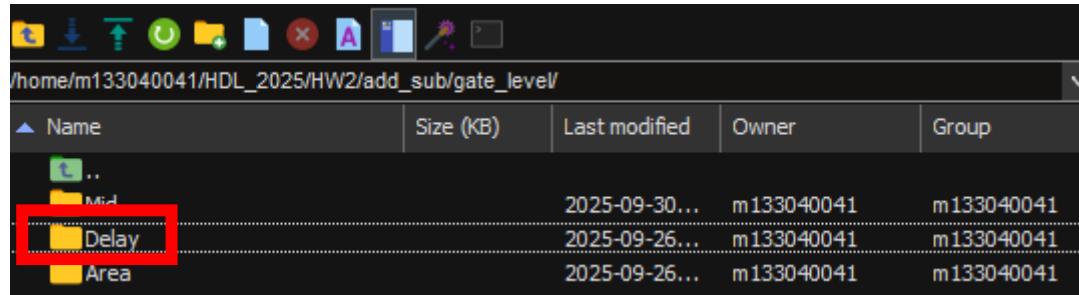
HW2資料夾建立規則

- 進入資料夾後 再建立這五個資料夾
- rtl:存放.v檔
- pt:存放primetime模擬資料，跑pt位置
- pre_sim:存放前測tb，跑前模擬位置
- post_sim:存放後側tb與，跑後模擬位置
- gate_level:存放合成後資料位置
- compile:裡面存放dc.tcl，做合成位置

	..		
	rtl	2025-09-25...	m133040041
	pt	2025-09-25...	m133040041
	pre_sim	2025-09-25...	m133040041
	post_sim	2025-09-25...	m133040041
	gate_level	2025-09-25...	m133040041
	compile	2025-09-25...	m133040041

HW2資料夾建立規則

- gate_level可以再建立Delay Mid Area 存放合成不同timing的gate_level檔案 (.ddc 、 .v 、 .sdf 、 .sdc)
- 這次只要做Delay即可



Outline

- Compare Multiply-Add & Adder/Subtractor
- Write Verilog Code
- Write Testbench
- Run Pre Synthesis Simulation – VCS
- Logic Synthesis – Design Compiler
- Run Post Synthesis Simulation – VCS
- PrimeTime
- Snapshot Waveform – nWave
- Complete Report

不同語法合成比較

- Multiply-Add Designs

```
module multiply_add (input [7:0] a, b, c, output [15:0] d);
assign d=a*c+b*c; // two 8x8 multipliers ; one 16-bit adder
endmodule
```

```
module multiply_add_common_factor (input [7:0] a, b, c, output [15:0] d);
assign d=(a+b)*c; // one 8-bit adder; one 8x8 multiplier;
endmodule
```

- Adder/Subtractor

```
module add_sub_1 (input [7:0] a, b, input s, output [15:0] d);
wire [7:0] tmp;
assign tmp = s ? b : -b; // one 2-to-1 multiplexer for possible negation of b
assign d = a + tmp; // an adder
endmodule
```

```
module add_sub_2 (input [7:0] a, b, input s, output [15:0] d);
assign d = s ? a+b : a-b; // one adder, one subtractor, one 2-to-1 multiplexer
endmodule
```

```
module add_sub_3 (input [7:0] a, b, input s, output reg [15:0] d);
always @ (a or b or s) // one adder, one subtractor, one 2-to-1 multiplexer
  if (s == 1) d = a+b; else d=a-b;
endmodule
```

不同語法合成比較

- 因為要在相同條件下比較

1. Multiply-Add Designs

2. Adder/Subtractor

- 所以clock都設定在1ns底下做合成就好
- (比較的RTL Code不用附上)

```
#Specify Clock , clock名需和top module中clk port相同
set Clk_pin          "clk"
set Clk_period        "1"
```

不同語法合成比較

- dc.tcl語法說明:

```
# Resource Sharing  
set hlo_resource_allocation area_only
```

- 設定為area_only是可以讓design compiler合成時自動做resource sharing(2024,2025DC不適用)

- 可以用start_gui 指令觀察

```
if (s) begin  
    result = a + b;  
end else begin  
    result = a + c;  
end
```

繳交檔案(繳交整包HW2即可)

- Testbench
- RTL
- Gate Level
- PT
- Vivado
- Report.pdf
- 以上只是評分會看的檔案，不須額外整理

繳交檔案

- Testbench
 - Clock_gating
 - pre_syn_testbench.v
 - post_syn_testbench.v
 - Non_pipeline
 - pre_syn_testbench.v
 - post_syn_testbench.v
 - pipeline
 - pre_syn_testbench.v
 - post_syn_testbench.v
- RTL
- Gate Level
- PT
- Vivado
- Report.pdf

Testbench

1. testbench的輸入a、b、c、s使用random產生，至少要200組(可以更多組)，而c必須要50%的數量為非0值、50%為0值，也就是TB中c有100個要非0值 有100個要0值。限制c的0與非0數量，在做pipeline與clock-gating的power比較較容易看出差異。
2. 此作業可不考慮singed、unsigned、以及溢位，或是也可以在TB上限制若在a-b時b的數值小於a。並且輸出d要保持在16位元
3. TB在做驗證時請使用\$display()或\$strobe()或\$monitor()等verilog語法去列印輸出與輸入值，可方便做驗證，在截圖時也請麻煩附上部分列印結果

繳交檔案

- Testbench
- RTL
 - Clock_gating
 - clockgating.v
 - Non_pipeline
 - nonpipeline.v
 - pipeline
 - pipeline.v
- Gate Level
- PT
- Vivado
- Report.pdf

RTL

1. 這次作業不需要跟作業1一樣做delay-optimization、area-optimization、in-between，**只需要做delay-optimization的合成就好**。
2. 在做pipeline、clock-gating各階的pipeline delay，每階的pipeline要分module寫，這樣在做DC的timing report 才能用unlevel找到各階module的delay。(指令可以看作業2的dc.tcl，並且要把註解拿掉)
3. Clock gating的寫法請使用 Clock Gating 2025.pptx第12頁方法2的推薦寫法(non-latch Version)，之後再用dc.tcl的replace_clock_gates轉成Latch-based Clock Gating。請別使用方法一的Auto Clock Gating方法。

繳交檔案

- Testbench
- RTL
- Gate Level
 - Clock_gating
 - delay
 - Non_pipeline
 - delay
 - pipeline
 - delay
- PT
- Vivado
- Report.pdf

繳交檔案

- Testbench
- RTL
- Gate Level
- PT
 - Clock_gating
 - report_power_average_vcd_hw2_gclk.rpt
 - Non_pipeline
 - report_power_average_vcd_hw2_nonpipe.rpt
 - pipeline
 - report_power_average_vcd_hw2_pipe.rpt
- Vivado
- Report.pdf

PT

1. 在比較primetime輸出的power上，一定要的看出來 clock-gating的total power會比pipeline小，並且在做 clock-gating與 pipeline的pattern要相同，並且也請查看Gating DFF使用比例是否有成功與達到預期的結果。(指令可以看作業2的dc.tcl或作業2的ppt，並且使用tcl時要把註解拿掉)
2. 作業2的primetime不需要做分開模擬c為0與c不為0的情況，也就是說vcd的波型檔一個電路只會有一個，例如作業有3個電路(non-pipelined、pipeline、clock-gating)，那只會有3個vcd檔案，並且每個vcd就記錄了c為0(100筆)與c不為0(100筆)的情況(共200筆)，讓TB直接跑完200筆資料就好。此作業也不用交vcd。
3. primetime的power結果也要截圖。

繳交檔案

- Testbench
- RTL
- Gate Level
- PT
- Vivado
 - Pipeline.xdc
 - Behavior.wcfg
 - post-imp.wcfg
 - .xpr.zip
- Report.pdf

Vivado

1. vivado 請用自己的，server的用不了。
2. vivado不需要合non-pipelined與clock-gating，只需要做pipeline即可。
3. 在做Post-imp -Simulation時，把reset 訊號拉到至少100ns時輸出結果才不會出錯。
4. 把Vivado的資料夾放進HW2裡面一起繳交

繳交檔案

- Testbench
- RTL
- Gate Level
- PT
- Vivado
- Report.pdf
 - 合成寫法比較與發現
 - 模擬波形圖
 - RTL level
 - Gate level (delay optimize)
 - 數據表格
 - 心得

Report.pdf

1. word檔放上Multiply-Add & Adder/Subtractor，Design compiler 合成delay, area and power 數據截圖，以及不同寫法合成的比較及發現。
2. word檔必須要有Pre-sim與 Gate-level-sim的波形截圖(3個電路*2)與解釋，Design compiler的area、power、timing截圖以及 pipeline、clock-gating的各階pipeline delay，部分驗證截圖(第9點)，Primetime power測量截圖，Vivado Behavior 與 post implement 波形和解釋，Vivado Summary Overview 截圖。
3. 作業2 PDF上的表格也要在呈現並填寫在Word裡。

配分

- Testbench(pre_sim&post_sim 10%)
- RTL(5%)
- Gate Level(ddc、.v、.sdf、.sdc 10%)
- PT (5%)
- Vivado(HDL_HW2_MXXXXXXXXXX.xpr.zip 15%
xdc、wcfg 2.5%)
- Report.pdf(52.5%)
- 請不要上傳製程檔案(N16ADFP_StdCell.v) , 上傳扣50分。

分配

- Report.pdf(52.5%)
- Design Complier(40%)
- 合成寫法比較截圖&發現 (5%)
- RTL波形 & gate-level波形並解釋 (5%)
- Design Compiler Delay, Power, Area, Prime Time測量截圖(10%)
- 測量二階不同pipeline內的 timing資訊(5%)
- 自動驗證運算結果 (5%)
- 數據表格(10%)
- Vivado(12.5%)
- Behavior波形 & post-implement波形並解釋 (10%)
- Project Summary-Overview截圖，含 Utilization 、 Timing(2.5%)

數據表格

	Area (μm^2)			Delay (ns)	Latency (ns)	Power (W)		
	CL	SL	Total			dynamic	leakage	total
Non-pipelined (DC)								
Non-pipelined (PrimeTime)								
Pipelined (DC)								
Pipelined (PtimeTime)								
Clock-gated (DC)								
Clock-gated (PrimeTime)								

作業繳交

- 1.直接複製整個HW1資料夾到Server內繳交(不須壓)
- 指令：

`cp -r /MasterClass/M133040041_HDL/HW1//MasterClass/Homework-Submit/M133040041_HDL/`
(請替換 M133040041_HDL 成自己的學號)

- 2上傳加入封面的Report.pdf到網大作業區

Due:11/2 23:59