

HDL Digital Design (Graduate Level)

Spring 2024

Final Bicubic REPORT

Must do self-checking before submission:

- ☒ Compress all files described in the problem into one **zip file**.
- ☒ All files can be compiled under **ModelSim & Quartus** environment.
- ☒ All port declarations comply with I/O port specifications.
- ☒ Organize files according to File Hierarchy Requirement
- ☒ No **waveform files or project file** in deliverables

Due Date: 2024/06/13 8:59:59 a.m.

Student name: ____林柏戎____

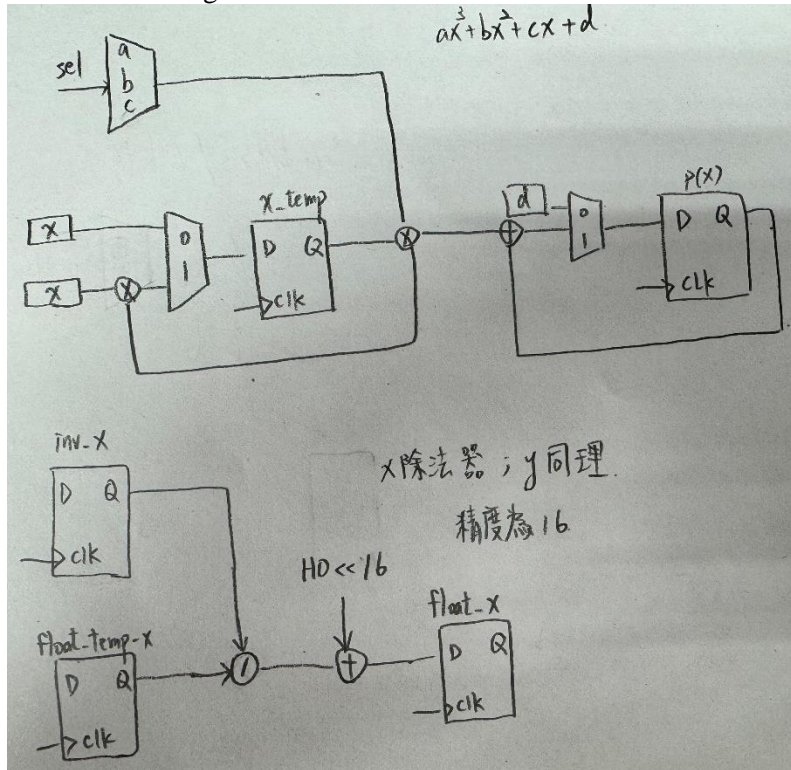
Student ID: __Q36114239__

1. Performance & Area table

Gate level simulation all pass (Yes/No)	Yes
Total error pixels	0
Clock period in testbench	54.55 (ns)
Simulation time in Gate Level	1523834 (ns)
Area (Total logic elements + 500 * embedded Multiplier)	2235

2. Architecture Diagram in this lab

(5%)



3. Explain algorithm & how your design works?

(25%)

IDLE: 初始化的狀態，FINISH 完，也會回到 IDLE 狀態。

CHECK: 決定要走 INIT_SEL 還是 SELECT。這個狀態會知道目前是否在第一橫排的 data。第一橫排預處理的 data 會比其他 pixel 來的多，主要是為了減少 MEM 要 data 的次數，這樣到下一橫排時，原本要讀 4 個 data 的，可以只讀 1 個；原本要讀 16 個 data 的，可以只讀 4 個。這樣可以大幅減少 cycle 數。

➤ **INIT_SEL:** 依照目前 x 的是否有小數點，來決定要走 INIT_READ 還是 INIT_INTER。

✓ **INIT_READ:** 只需要讀 4 個 data 就好，這主要在處理完全重合的 case，所以會接去 WRITE 狀態，寫入第 0 個 pixel 的 data 是多少。

- ✓ **INIT_INTER**: 需要讀 12 個 data，這主要在預處理需要計算 5 次 interpolation 的 case，因為只讀 12 個 data，等於預處理 3 次 interpolation 的 data，之後也會接去 WRITE 狀態，寫入第 1 個 pixel 的 data 是多少。
- **SELECT**: 依照目前 {y, x} 是否有小數點，來決定要走 INTER_Y、INTER_X、INTER_XY。
- ✓ **INTER_Y**: 讀 1 個 data + 做一次 interpolation，這代表做直的 interpolation。會再接到 WRITE 狀態，將對應的 pixel addr 寫入對應的 pixel data。
- ✓ **INTER_X**: 因為前面的預處理已將橫排的 interpolation 做完，所以可以直接接到 WRITE 狀態，將對應的 pixel addr 寫入對應的 pixel data。
- ✓ **INTER_XY**: 讀 4 個 data + 做兩次 interpolation，配合前面已預處理好的 3 次 interpolation 的 data，這代表只需要再做一次橫的 + 一次直的 interpolation。會再接到 WRITE 狀態，將對應的 pixel addr 寫入對應的 pixel data。
- WRITE**: 將對應的 pixel addr 寫入對應的 pixel data。
- FINISH**: 將 DONE signal 拉高。並回到 IDLE 狀態，等待下一個 pattern。

4. RTL & Gate-level simulation result on the terminal.

(5%)

RTL:

```

*****
** Simulation Start          **
*****

== PATTERN pattern1
== PATTERN pattern2
== PATTERN pattern3

*****
**   Finish Simulation      **
** Error pixels:           0  **
** Simulation time: 1523834.00 ns **
*****

```

```

Loading instances from Bicubic_7_1200mv_85c_v_slow.sdo
Loading altera_ver.PRIM_GDFF_LOW
Loading timing data from Bicubic_7_1200mv_85c_v_slow.sdo
** Note: (vsim-3587) SDF Backannotation Successfully Completed.
Time: 0 ps Iteration: 0 Instance: /tb File: C:/Users/MediaCore/Desktop/StudentID_final_v2/tb.sv

add wave *
view structure
.main_pane.structure.interior.cs.body.struct
view signals
.main_pane.objects.interior.cs.body.tree
run -all

*****
** Simulation Start **
*****

== PATTERN pattern1
== PATTERN pattern2
== PATTERN pattern3

*****
** Finish Simulation
** Error pixels: 0
** Simulation time: 1523834.00 ns
*****

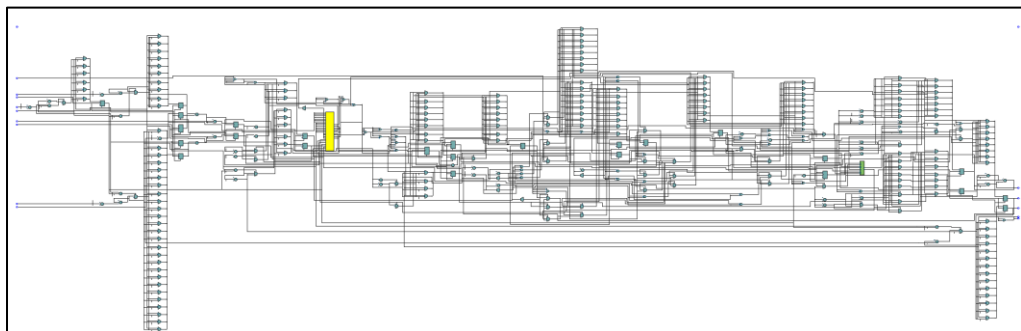
```

Gate-Level:

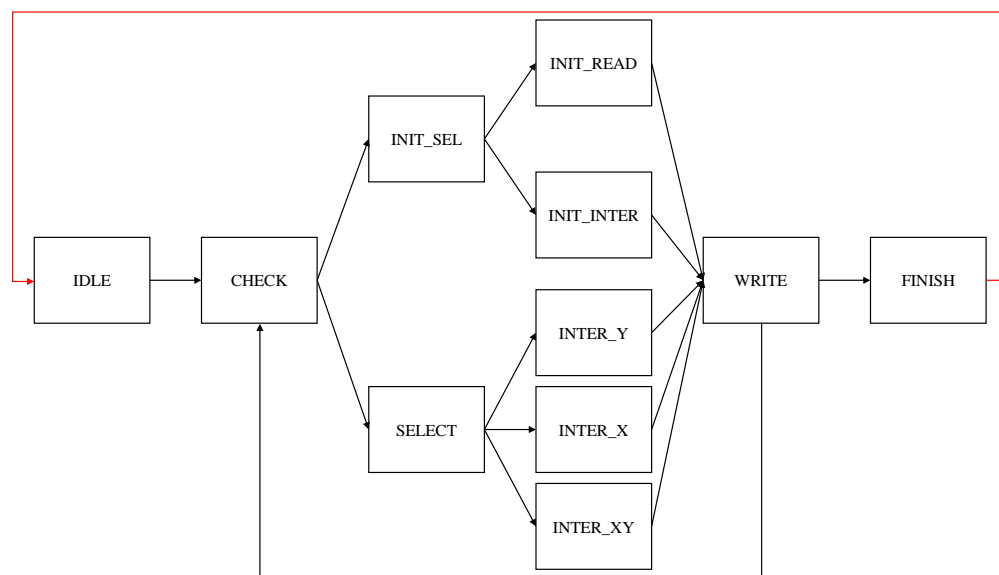
5. Screenshot form Quartus

(10%, 2% each)

- ✓ Schematic view of the design netlist



- ✓ Finite State Machine view



✓ Fmax summery

	Fmax	Restricted Fmax	Clock Name	Note
1	18.37 MHz	18.37 MHz	CLK	

✓ worst-case timing paths

	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	0.121	float_temp_X3[4]	float_x[0]	CLK	CLK	54.550	0.330	54.777
2	0.379	float_temp_Y3[1]	float_y[0]	CLK	CLK	54.550	0.333	54.522
3	0.419	inv_y[24]	float_y[0]	CLK	CLK	54.550	0.332	54.481
4	0.443	float_temp_X3[5]	float_x[0]	CLK	CLK	54.550	0.330	54.455
5	0.457	float_temp_Y3[5]	float_y[0]	CLK	CLK	54.550	0.333	54.444
6	0.460	float_temp_X3[1]	float_x[0]	CLK	CLK	54.550	0.331	54.439
7	0.507	float_temp_Y3[0]	float_y[0]	CLK	CLK	54.550	0.333	54.394
8	0.529	float_temp_X3[0]	float_x[0]	CLK	CLK	54.550	0.331	54.370
9	0.542	float_temp_X3[3]	float_x[0]	CLK	CLK	54.550	0.330	54.356
10	0.637	float_temp_Y3[4]	float_y[0]	CLK	CLK	54.550	0.334	54.265

✓ flow summery

Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	2,235 / 114,480 (2 %)
Total registers	317
Total pins	86 / 529 (16 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

6. Please explain what setup time and hold time violations are , and how you avoided or resolved them in this semester's labs? (25%)

➤ Setup time violation: 當 flip-flop 的 data 輸入變化太接近 clk 而不滿足最小 setup time 的要求時，就會發生 setup time violations。這代表 flip-flop 可能無法接受到正確數據，進而可能發生 metastability。

✓ **增加 Clk Period:** 降低時脈頻率、增加時脈週期，可能可以為資料在下一個 clk edge 來之前穩定，提供更多時間。

✓ **Pipeline Stages:** 新增 Pipeline Stages 可以將較長的 combinational logic path 分解為較小的段，減少資料傳播通過每個 stage 所需的時間。

- ✓ **Optimize 邏輯**：簡化或 Optimize 組合邏輯，以減少資料路徑中的延遲。
- ✓ Clock Tree Synthesis (CTS): 優化 clk distribution 以最大限度地減少偏差並確保時脈訊號同時到達不同的 flip-flops。
- ✓ 使用 Timing Constraints: 在合成或是 P&R 期間使用適當的 timing constraints，以確保 design 滿足 setup time 要求。
- Hold time violation: 當輸入到 flip-flop 的 data 在 posedge clk “後”，太快改變而不滿足最小 hold time 的要求時，就會發生 hold time violations。這可能導致 flip-flop 會寄存到錯誤的 data，亦可能發生 metastability。
- ✓ **增加 Data Path Delay**: 在 Data Path 中加入少量延遲(例如額外的 logic 或 buffers)，確保資料不會在 clk edge 過快改變。
- ✓ Clock Tree Synthesis (CTS): 再優化 clk distribution 以減少偏差。這確保 hold time signal 不會太早到達某些 flip-flops。
- ✓ 使用時序約束: 與 setup time 類似，使用適當的 timing constraints。

7. lesson learned in this lab and this semester. (30%)

- 了解和實現一維和二維的 Bicubic 插值算法，這是一種常用的 image resize 技術，能夠在放大圖像時保持 image 的細節。
- 熟悉和掌握使用 Verilog 進行硬體描述和設計，特別是針對影像處理電路的設計。之前是實現 image convolution，這次是 image resize。
- 學習到如何在實現記憶體讀取/寫入操作圖像數據，理解一維和二維數據的轉換和存取方式。
- 將 code 寫成巨集的方式，一開始不知道精度要設定為多少，所以在最上面設定一個 define 來控制精度。
- 在 FPGA 上實現除法器，若只用"/"，critical path 會很長，大概約 50 左右。如果自己設計除法器，要使用 pipeline 的方式來 design，critical path 會縮短很多，但會多出 "整數 bits+小數 bits+3" 個 cycle 左右。
- 可以嘗試在最前面用自己設計的除法器，後去只需要做累加的計算，再針對整除的部分做處理。在 critical path 會降低很多，雖然 total cycle 會多一些，但因為 critical path 降很多，所以整體的 PA 應該會降很多。