2017 Digital IC Design
Homework 4: Edge-Based Line Average interpolation

1. Introduction

The interlaced video comprises two types of fields in the sequence, one is the odd and another is the even field. The de-interlacing process is to convert the interlaced video into the non-interlaced form as shown in Fig. 1. The simplest method is intra-field interpolation, which use the existing pixels in the field to generate the empty lines. For instance, the empty lines can be filled via line doubling, which is quite easy to be implemented but the resulting image is not good enough in visual quality. In this homework, you are asked to implement the Edge-Based Line Average interpolation algorithm. As the direction of edge is considered, the de-interlaced image has a better quality than merely doubling the existing lines.

(a) (b)

Fig. 1. The odd field of an interlaced video sequence (a) and the complete frame after de-interlacing (b).

Assume that the pixel to be interpolated is located at coordinate (i , j) and pixels *a* to *f* are the neighboring points, which is shown in Fig. 2. First of all, three different directions at the interpolated position is calculated using (1), and the value of interpolated pixel is obtained by (2) and output before rounding.

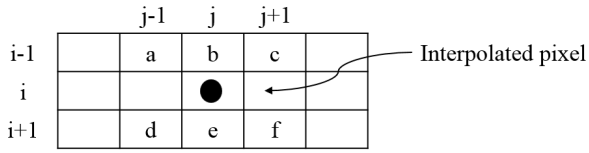


Fig. 2. The interpolated pixel and its neighboring points

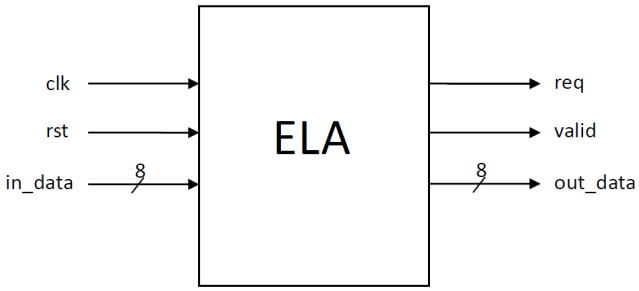
$$\begin{cases} D_1 = |a - f| \\ D_2 = |b - e| \\ D_3 = |c - d| \end{cases} \quad \text{..... (1)}$$

$$x(i,j) = \begin{cases} (a+f)/2, \text{if } \min(D_1, D_2, D_3) = D_1 \\ (b+e)/2, \text{if } \min(D_1, D_2, D_3) = D_2 \\ (c+d)/2, \text{if } \min(D_1, D_2, D_3) = D_3 \end{cases}, \quad \text{..... (2)}$$

If there are identical direction values, the priority of the three directions is $D_2 > D_1 > D_3$. For instance, $D_1=20$, $D_2=30$, $D_3=20$, then $\min(D_1, D_2, D_3) = D_1$. The left and right boundary interpolation is fixed to (b+e)/2.

2. Design Specifications

2.1 Block Overview



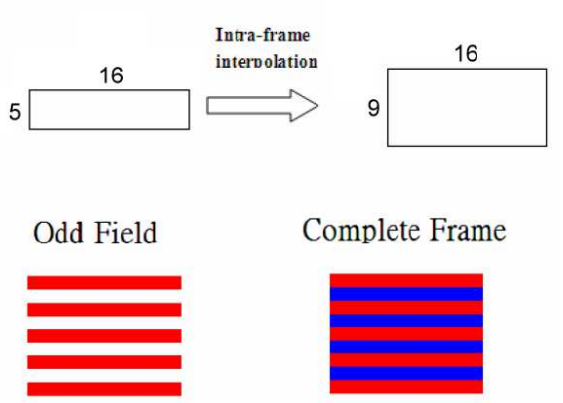
2.2 I/O Interface

Signal Name	I/O	width	Description
clk	I	1	clock for the ELA system
rst	I	1	active high synchronous reset
in_data	I	8	The 8-bit input pixel data.
req	O	1	The request signal for a line of pixels.
valid	O	1	Output signal. The output data is valid when the signal is high. The req signal must be 0 when you set valid to 1.
out_data	O	8	The 8-bit output pixel data.

2.3 File Description

File Name	Description
ELA.v	RTL code for using Verilog
testfixture.v	Test bench for verifying design
test1.data	Input pattern 1
test2.data	Input pattern 2
golden1.data	Golden output pattern 1
golden2.data	Golden output pattern 2
cycloneii_atoms.v	Simulation library for gate-level simulation

The given test patterns are 16x5 image, you have to output the complete 16x9 image!



3. Scoring

3.1 Functional Simulation (pre-sim) **[60%]**

All of the result should be generated correctly using two test patterns (modify the testfixture), and you will get the following message in ModleSim simulation. You can turn off the timing check in pre-sim only.

```
VSIM 10> run -all
# ----------------------------------------------------------
# \^0^/ All data have been generated successfully! \^0^/
# ----------------------------------------------------------
# ** Note: $finish    : E:/2016DICHW/HW4/testfixture.v(67)
#    Time: 7025 ns  Iteration: 0  Instance: /test
# 1
# Break in Module test at E:/2016DICHW/HW4/testfixture.v line 67
```

3.2 Gate-Level Simulation (post-sim) **[20%]**

3.2.1 Synthesis

Your code should be synthesizable. After synthesizing in Quartus, the file named *ELA.vo* and *ELA.sdo* will be obtained.

Device : Cyclone II EP2C70F896C8

3.2.2 Simulation

All of the result should be generated correctly using two test patterns by *ELA.vo* and *ELA.sdo*, and you will get the pass message in ModleSim simulation. (There should be no setup or hold time violations.)

3.3 Performance **[20%]**

The performance is scored by the logic elements you used and the simulation time in post-sim. The scoring equation is *(Total logic elements + total memory bit + 9*embedded multiplier 9-bit element) × (longest gate-level simulation time in ns)*. (The smaller the better).

Flow Status	Successful - Thu Nov 17 21:39:38 2016
Quartus II Version	10.0 Build 262 08/18/2010 SP 1 SJ Full Version
Revision Name	ELA
Top-level Entity Name	ELA
Family	Cyclone II
Device	EP2C70F896C8
Timing Models	Final
Met timing requirements	Yes
Total logic elements	459 / 68,416 (< 1 %)
Total combinational functions	427 / 68,416 (< 1 %)
Dedicated logic registers	159 / 68,416 (< 1 %)
Total registers	159
Total pins	20 / 622 (3 %)
Total virtual pins	0
Total memory bits	0 / 1,152,000 (0 %)
Embedded Multiplier 9-bit elements	0 / 300 (0 %)
Total PLLs	0 / 4 (0 %)

```
/*Modify by yourself*/
`define CYCLE 22
`define TEST_DATA "test2.data"
`define GOLDEN_DATA "golden2.data"
/*********************/
```

```
# ----------------------------------------------------------
# \^0^/ All data have been generated successfully! \^0^/
# ----------------------------------------------------------
# ** Note: $finish    : G:/2016DICHW/HW4/simulation/modelsim/testfixture.v(67)
#    Time: 3844 ns  Iteration: 0  Instance: /test
# 1
# Break in Module test at G:/2016DICHW/HW4/simulation/modelsim/testfixture.v line 67
```

4. Submission

4.1 Submitted files

You should classify your files into three directories and compressed to .zip format. The naming rule is **HW4_*studentID_name_version*.zip**. The *vision* is v1 for the first submission, and v2, v3… for the revisions.

RTL category	
*.v	All of your verilog RTL code
Gate-Level category	
*.vo	Gate-Level netlist generated by Quartus
*.sdo	SDF timing information generated by Quartus
Documentary category	
*.pdf	The report file of your design (in pdf).

4.2 Report file

You have to describe how the circuit is designed as detailed as possible, and the flow summary result and simulation results are necessary. **Please follow the specification in appendix.**

4.3 Please submit your .zip file to folder HW4 in the ftp site.

Deadline: 2017-12-12 23:59

ftp : 140.116.245.92

Usermame : ic_design

Password : icdesign

5. If you have any problem, please contact the TA by email:

weiting84610@gmail.com

p78031175@mail.ncku.edu.tw

2017 Digital IC Design

Homework 4: Edge-Based Line Average interpolation

NAME	
Student ID	

Simulation Result

Test pattern 1	Functional simulation	Pass or Fail	Gate-level simulation	Pass or Fail	Gate-level simulation time	simulation time (ns)
Test pattern 2	Functional simulation	Pass or Fail	Gate-level simulation	Pass or Fail	Gate-level simulation time	simulation time (ns)

your pre-sim result of two test patterns	your post-sim result of two test patterns

Synthesis Result

Total logic elements	
Total memory bit	
Embedded multiplier 9-bit element	

(your flow summary)

Description of your design

*Scoring = (Total logic elements + total memory bit + 9*embedded multiplier 9-bit element) × (longest gate-level simulation time in ns)*