

Computer-Aided VLSI System Design, Fall 2011

Verilog HW 3

1. Design Overview

In this homework, we will design a Local Median Filter Engine (LMFE). Fig. 1 shows the block diagram of this LMFE. The signal names are also shown in the figure. The main idea of the median filter is to run through the signal pixel by pixel. Then the median filter outputs and replaces the each pixel with the median of neighboring pixels.

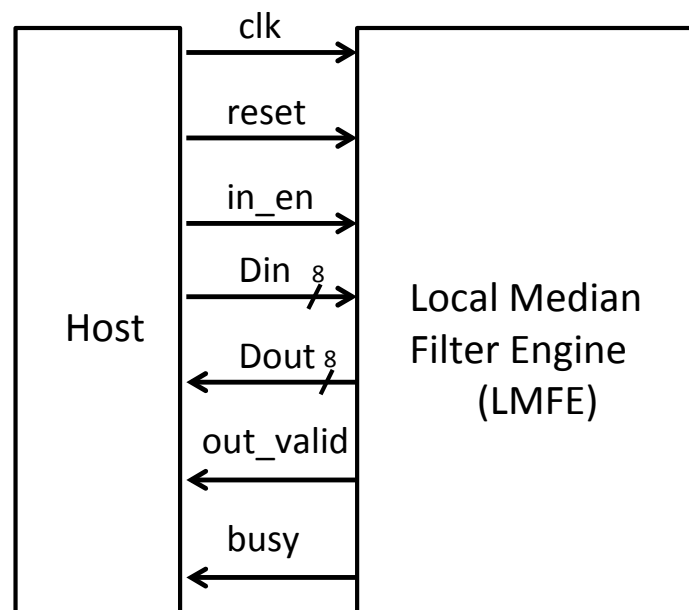


Figure 1. Block diagram of LMFE

2. Specification

2.1 Input Signals and Output Signals

Signal Name	I/O	Width	Simple Description
clk	I	1	The clock of LMFE, all signal timing are related to the rising edge of clk.
reset	I	1	The reset is an active high asynchronous signal.
in_en	I	1	Input data enable signal: in_en=0(disable) in_en=1(enable)
Din	I	8	input data(one pixel pre cycle)

busy	O	1	The busy signal of LMFE: busy=0 (Host can input data) busy=1 (Host can not input data)
out_valid	O	1	Output control signal: out_valid=0 (invalid output data) out_valid=1 (valid output data)
Dout	O	8	output data(one pixel pre cycle)

2.2 Description of System

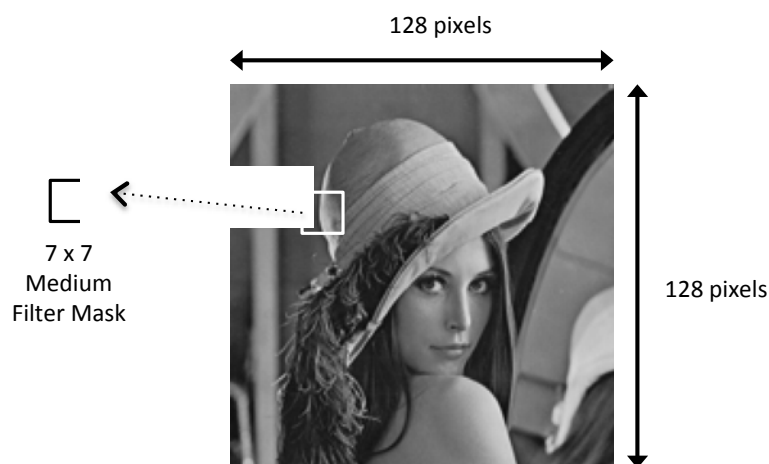


Figure 2. Function of LMFE

2.2.1 Image Input

Input data is a gray-level 128 x 128 image, which totally has 16384 pixels, and each pixel is 8-bit data (0~255), as shown in Fig. 2. We use the raster scan to read image, which is a line-by-line scanning and shown in Fig. 3 (sequence of data: 0,1,2,4 ...16383).

<Note>

- You only have one chance to read a pixel.
- If stopping the input of image, you can assert the signal “busy” to High (logic 1).

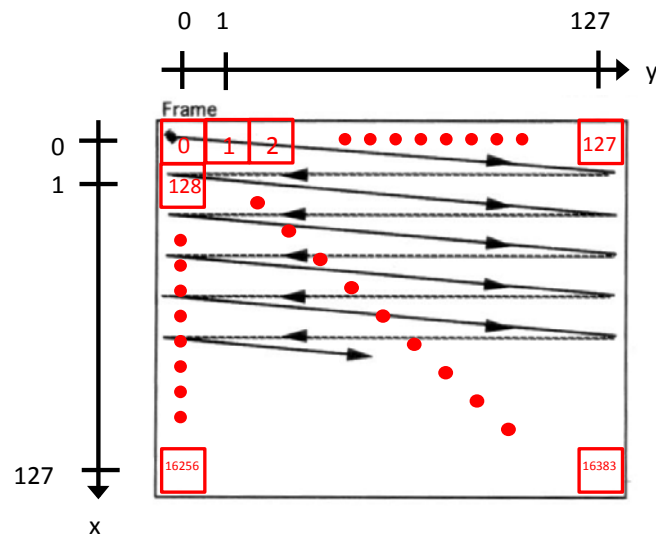


Figure 3. Image Input

2.2.2 Median Filter

This LMFE is a 7 x 7 median filter, which finds the median value in this 7 x 7 mask. Then the LMFE will replace original pixel with the median value of the 7 x 7 mask. Two examples of the 3 x 3 median filter are shown as below:

(1) General Point

Example: 3 x 3 mask, pixel (2,2)

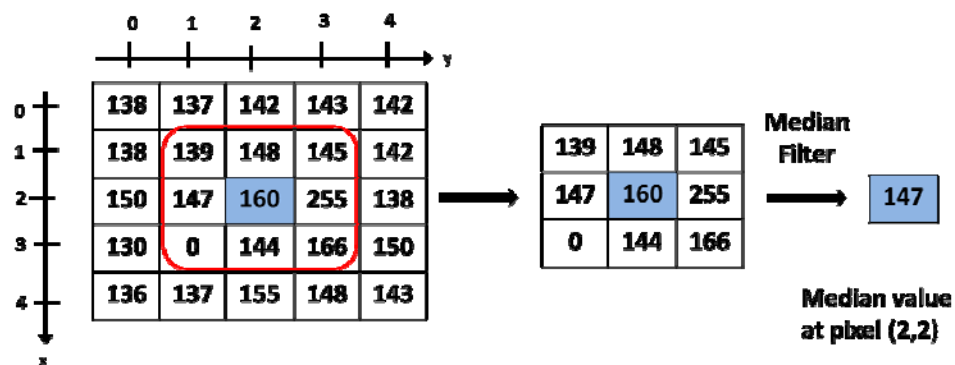


Figure 4. Example of 3 x 3 LMFE for (2,2)

(2) Boundary Point

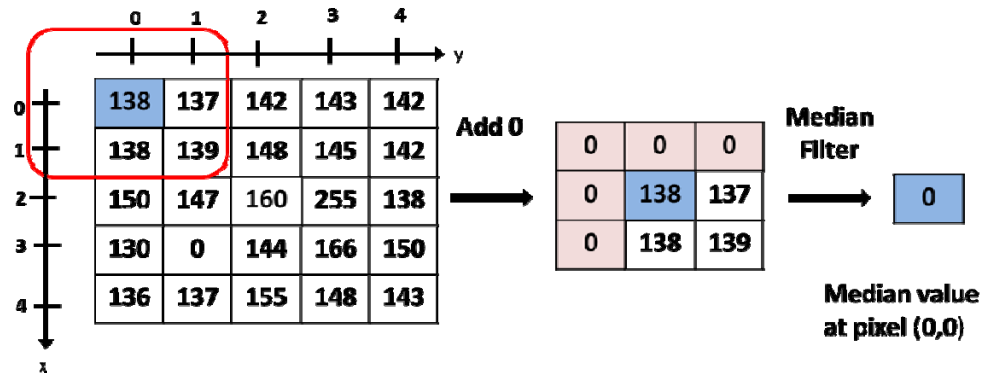
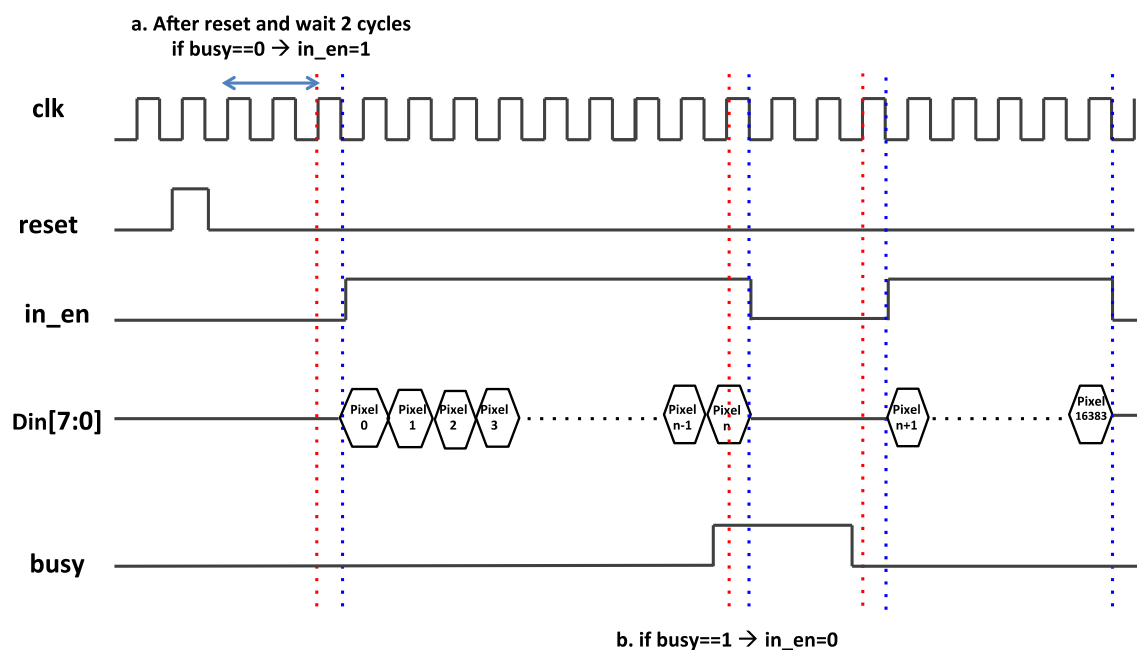
Example: 3 x 3 mask, pixel (0,0)

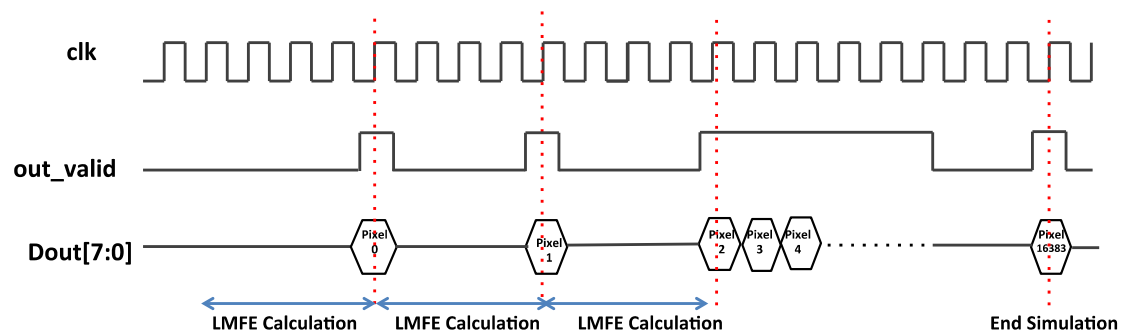
Figure 5. Example of 3 x 3 LMFE for (0,0)

2.2.3 Image Output

Output sequence is also shown in Fig.3. (0,1, 2, 3, ...,16383). If output data is valid, the signal “out_valid” need to be set to high (logic 1).

2.3 Timing Diagram**2.3.1 Input Signal**

2.3.1 Output Signal



2.4 SRAM Specifications and Timing Diagram

In this homework, you **should** use the single port SRAM to buffer pixels. This SRAM operation has been introduced in Lab2. You need to generate the SRAM by using the memory compiler. The command of memory compiler is shown as follow:

```
~cvstd/CBDK_IC_Constest/CIC/Memory/ralshd/bin/ralshd &
```

(Note 1: it is a single-line command!)

(Note 2: when typing a long path, try TAB key!)

Two Kinds of SRAM Specification: (depend on your design)

1. 1KB (1024 x 8bit) SRAM

Instance name: *sram_1024x8_t13*

Number of words: 1024

Number of bits: 8

Also, you need to generate worst-case library (*sram_1024x8_t13_slow_syn.lib*) for use in Synopsys Synthesis tool.

2. 8KB (8192 x 8bit) SRAM

Instance name: *sram_8192x8_t13*

Number of words: 8192

Number of bits: 8

Also, you need to generate worst-case library (*sram_8192x8_t13_slow_syn.lib*) for use in Synopsys Synthesis tool.

For example: If you want to use a 16KB SRAM, you can use **2 8KB SRAMs** or **16 1KB SRAMs**

3. Grading Policy

a. (30%) RTL correctness (LMFE.v)

b. (70%) Gate-level simulation correctness (LMFE_syn.v): without any problem of setup time & hold time. Please follow the Design Vision setting rules in the next section.

The evaluation formula is

$$\text{Score} = \text{Area}(\text{um}^2) * \text{Simulation time (ns)}$$

where the Area is the “total area (um²)” in your Design Vision area report, and the simulation time (ns) is the “Gate-level Pre-layout Simulation Time”. The example of simulation time is shown as below.

The example of simulation time: (**Simulation time=167761 ns**)

```
-----
Congratulation! All data have been generated successfully!
-----PASS-----
Simulation complete via $finish(1) at time 167761 NS + 0
./testfixture1.v:135      #(`CYCLE/2); $finish;
```

A Class: ~5% (70 points)

B Class: 6%~ 20% (60 points)

C Class: 21%~70% (50 points)

D Class: 71%~100% (40 points)

E Class: There are some errors in the gate-level simulation. (0 point)

Any cheat will make your homework failed.

<Note>

a. Synthesis:

Environment Setup: please refer the file “synoposys_dc.setup” that is given by synthesis lab.

target_library	slow.db, sram_8192x8_t13_slow_syn.db (or sram_1024x8_t13_slow_syn.db)
----------------	--

link_library	* \$target_library dw_foundation.sldb
symbol_library	generic.sdb
synthetic_library	dw_foundation.sldb

Design constraints: **please source “LMFE_DC.sdc”**

b. In gate-level simulation, you have to use the following command:

➤ ncverilog testfixture1.v LMFE_syn.v -v tsmc13.v +define+SDF +access+r

For gate-level simulation, remember to modify the parameter “CYCLE” to your target clock period. Also, remember to modify the parameter “SDFFILE” to the name of your SDF file.

4. Submission

RTL: *LMFE.v*

Synthesis: *LMFE_syn.v*

Memory: *Verilog Model & Synopsys Model*

Report: **pdf file including Area Report & Gate-level Pre-layout Simulation Time**

Please submit your design in one file with the naming convention: *StudentID_HW3_vk.zip* (for version $k = 1, 2, \dots$). Please follow the electronic submission guideline (available on website).

5. Description of Files

File name	Description
LMFE.v	Design file
testfixture1.v testfixture2.v	Test bench 1 & 2. Modify the parameter “CYCLE” to meet your requirement of clock period.
pattern_1.dat pattern_2.dat	Test pattern 1 & 2. Each test pattern has 16384 pixels.
golden_1.dat golden_2.dat	Golden pattern 1 & 2.
LMFE_DC.sdc	Design constrains for synthesis.