

EE6470 Electronic System Level Design and Synthesis

Homework4 Report

108062209 王鴻昱

- Github repo: <https://github.com/PaulWang0513/Electronic-System-Level-Design-and-Synthesis>

- Problem Description and Solution

In this homework, we are asked to port the Median&MeanFilter module to the “basic-acc” platform in Lab8. So, I first reused the SobelFilter module framework to build a MedianFilter module. Then I imitated the structure of the module to write code to expand the MedianFilter module into Median&MeanFilter module.

- Implementation Details

First, I reused the code of SobelFilter to build a MedianFilter, and learn how it works at the same time. Specifically, I modified 4 files from lab8, which are filter_def.h, SobelFilter.h, basic-acc/main.cpp, and basic-sobel/main.cpp.

In filter_def.h, I modified the name of SOBEL_FILTER_R_ADDR and SOBEL_FILTER_RESULT_ADDR into MEDIAN_FILTER_R_ADDR and MEDIAN_FILTER_RESULT_ADDR respectively, to make the code easy to read.

```
17 // Used between blocking_transport() & do_filter()
18- const int SOBEL_FILTER_R_ADDR = 0x00000000;
19- const int SOBEL_FILTER_RESULT_ADDR = 0x00000004;
```

```
15 // Used between blocking_transport() & do_filter()
→ 16+ const int MEDIAN_FILTER_R_ADDR = 0x00000000;
17+ const int MEDIAN_FILTER_RESULT_ADDR = 0x00000004;
```

In SobelFilter.h, I modified the name of everything with SOBEL to MEDIAN, and modified the code in do_filter() to make it compute the median from input 9 pixels.

In basic-acc/main.cpp, I re-define the accelerator name to median_mean_filter, but not sobel_filter, so I won't be confused.

In basic-sobel/main.cpp, I modified the code to change the order of input feeding and bitmap writing. By doing so, the output image won't be rotated.

So far, the MedianFilter module can work well on the vp platform.

After I figure out how the DMA interface work, I can add the mean filter function into the Median&MeanFilter module.

In filter_def.h, I add the transport address of mean filter to allow the DMA finding the proper address.

```
14 // MedianMean Filter inner transport addresses
15 // Used between blocking_transport() & do_filter()
16 const int MEDIAN_FILTER_R_ADDR = 0x00000000;
17 const int MEDIAN_FILTER_RESULT_ADDR = 0x00000004;
18 const int MEAN_FILTER_R_ADDR = 0x00000008;
19 const int MEAN_FILTER_RESULT_ADDR = 0x0000000C;
```

In MedianMeanFilter.h, I add extra channels and blocking transport function's case for mean filter's data transport. And, of course, add the do_mean_filter() thread to compute the mean value from input data.

```
sc_fifo<unsigned char> i_r_median;
sc_fifo<unsigned char> i_g_median;
sc_fifo<unsigned char> i_b_median;
sc_fifo<int> o_result_median;
sc_fifo<unsigned char> i_r_mean;
sc_fifo<unsigned char> i_g_mean;
sc_fifo<unsigned char> i_b_mean;
sc_fifo<int> o_result_mean;
```

```
case tlm::TLM_READ_COMMAND:
// cout << "READ" << endl;
switch (addr) {
case MEDIAN_FILTER_RESULT_ADDR:
buffer.uint = o_result_median.read();
break;
case MEAN_FILTER_RESULT_ADDR:
buffer.uint = o_result_mean.read();
break;
```

```
case tlm::TLM_WRITE_COMMAND:
// cout << "WRITE" << endl;
switch (addr) {
case MEDIAN_FILTER_R_ADDR:
i_r_median.write(data_ptr[0]);
i_g_median.write(data_ptr[1]);
i_b_median.write(data_ptr[2]);
break;
case MEAN_FILTER_R_ADDR:
i_r_mean.write(data_ptr[0]);
i_g_mean.write(data_ptr[1]);
i_b_mean.write(data_ptr[2]);
break;
```

Nothing changes in basic-acc/main.cpp, because we only do the module instantiate and connect with bus here.

In In basic-sobel/main.cpp, I add codes to feed data to the mean filter by asking DMA to transport data in between specific address.

```
// MedianMean Filter ACC
static char* const MEDIANFILTER_START_ADDR = reinterpret_cast<char* const>(0x73000000);
static char* const MEDIANFILTER_READ_ADDR = reinterpret_cast<char* const>(0x73000004);
static char* const MEANFILTER_START_ADDR = reinterpret_cast<char* const>(0x73000008);
static char* const MEANFILTER_READ_ADDR = reinterpret_cast<char* const>(0x7300000C);
```

After these modifications, the MedianMeanFilter can run smoothly to process the noised image.

- Additional Features

No additional features this time.

- Experiment Results

No experiment required this time. The screenshot is part of the simulation result.

```
mean, height: 504
mean, height: 505
mean, height: 506
mean, height: 507
mean, height: 508
mean, height: 509
mean, height: 510
mean, height: 511
[sys_open] ./images/lake_noise_out.bmp, 1537 (translated to 577), 438

Info: /OSCI/SystemC: Simulation stopped by user.
=[ core : 0 ]=====
simulation time: 28468749880 ns
```

- Discussion and Conclusions

In this assignment, I learned the way to run accelerator in riscv-vp platform with DMA. I believe this can provides a lot of help when doing final project.