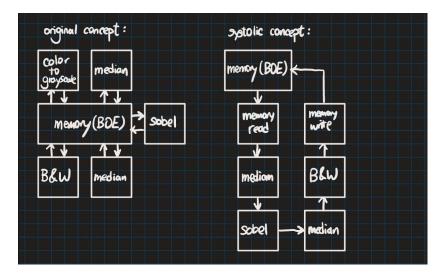
Monthly Report for ECE FYP/FYT

Project Code:	<u>ZW01a-23</u>	Supervisor(s):	Prof. Wei Zhang
Project Title:	Hardware Acceleration of Data Processing		
Group Member(s):	1) Luk Pak Him 2) Chio Y	at Hei	3) Shum Kwan Ho
Reporting Period:	Report #1		
Progress Report: List the work completed in this reporting period. Identify the major difficulties encountered. Comment on the overall progress.	Completed work: Median Filter Finalized design of systolic median filter (Issac) 3-sort module: sorting 3 8-bit numbers with 2 pipeline stages (Gordon) 9-median systolic: input 3 new numbers and compute the new median every clock cycle (Gordon) Sobel Filter Finalized design of systolic Sobel filter (Issac) Sobel module: single module, scalable for Sobel filter (Issac and Haven) Verify correctness by placing 3 Sobel filter CU side by side (Haven) Major difficulties: Figuring out the memory access of FPGA boards Group members are not all proficient in Verilog programming and HDL. Overall Progress: Slightly behind schedule, still haven't implemented the memory access part		
Future Plan: • Write down the working plan for the next reporting period.	 Implement scalable RTL schematic with for loops Implement memory access of BOE Implement Image division hardware 		
Group Representative's Signature:	In the	<u> </u>	MIV

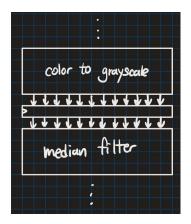
(Version 2020-10)

Appendix: Systolic Based Design

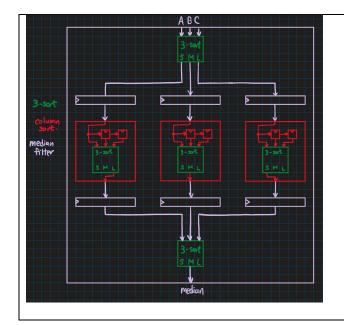


An image of width W and height H can be partitioned into multiple arrays of same height, but a smaller width to be fitted in the circuit.

Each clock cycle, a row of pixels is being loaded into the first step of the data processor, and the computational results would be stored intermediately in registers, waiting for the next task.



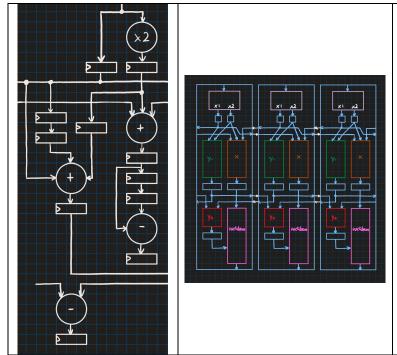
Design of module nine_median_systolic, a 9 8-bit median filter with sequential input and internal register



The design of a median filter module, as defined as a computational unit. Each CU handles one pixel output of median filter. The Filter takes in three new numbers in the same row for every clock cycle, and computes the output median using the values from the previous two clock cycles, as well as this cycle's value. It can be treated as a simple mealy machine.

three_sort is the module being implemented form the last progress report, with a major change in adding a additional pipeline stage in the middle between the comparators and MUXs.

The computational stage can simply be scaled up by adding more modules to run in parallel. Additional module would be made to route data into the median filter from previous stage.



The left side shows the detail of what to be implemented in a Sobel filter module. Just like the median filter module created above, the module can be treated as a complicated Moore machine. For each clock cycle, it takes in one pixel, and outputs one pixel.

Note that the module of Sobel filter cannot work alone, with the need of at least three to work. The right side shows three sobel_filter module placed side by side, with the internal modules also shown. The whole module is being split into multiple sub-modules for division of labor. The interconnects between sobel_filter modules are also modules called sobel_interconnect, which connects neighboring modules and exchange data to compute the final output.

The color to grayscale filter is obviously easier to implement, as it is a point operation. We are still designing how the black-and-white filter is to be implemented, as it is a global operation, where data might need to store back to the memory for further processing.