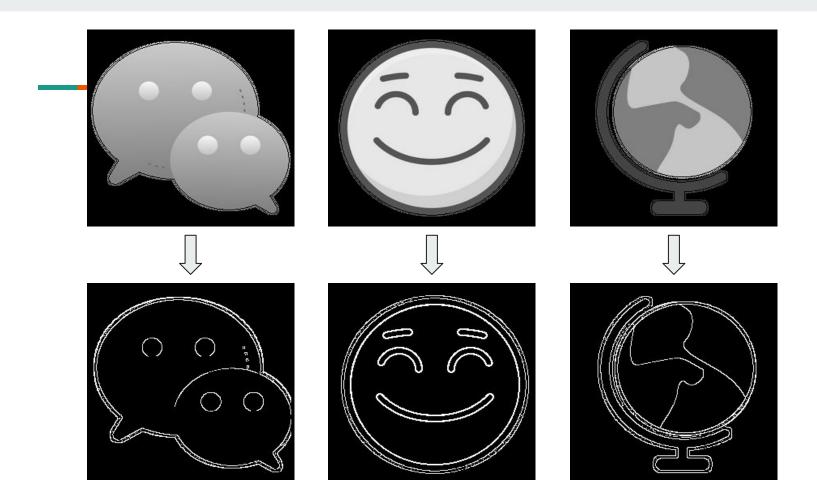
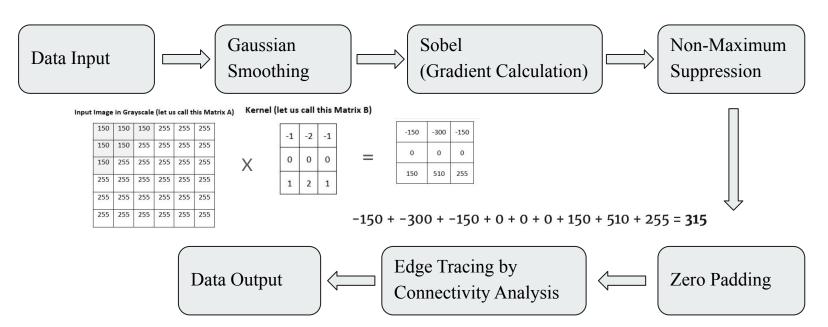
FPGA Based Acceleration of Canny Edge Detection

Design Overview

Our ultimate goal is to develop an Canny edge detection project that is not only optimized for performance but also implemented on the Zybo Z7-20 FPGA board.



Background(Canny Edge Detection)



Implementation

Input Helper Function

- 1. Read a 256×256 RGB image
- 2. Convert RGB image into Grayscale image using OpenCV
- 3. Generate .dat file containing each pixel value of the grayscale image

Input of Edge Detection Algorithm

- 1. Read .dat file to store the pixel values into array
- 2. Define HLS input stream with data type of bit32_t
- 3. Go through the pixel values in the array, pack each four pixels as a group to be written into the input stream

Edge Detection Process

- 1. Define HLS output stream with data type of bit32 t
- 2. Conduct edge detection using data from HLS input stream while store the output into the HLS output stream

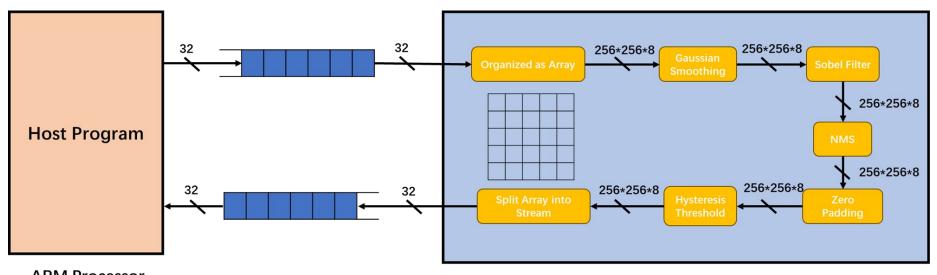
Output Helper Function

- 1. Read the .dat output file
- 2. Using OpenCV to restore the edge detected image according to the pixel values from output file.

Output of Edge Detection Algorithm

- Go through the data from output stream with each time retrieving four pixels as a group
- 2. Store the output pixel values into a .dat output file

Baseline Design



ARM Processor

Accelerator

Optimization Strategy

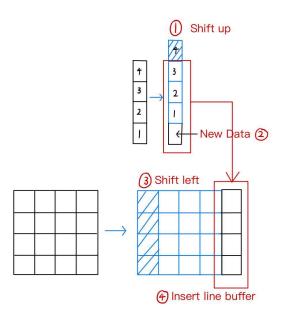
Abundant Parallelism:

- Unroll
- Pipeline
- Reshape
- Partition

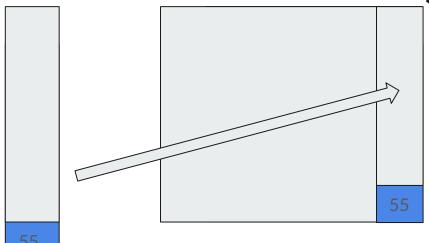
Distributed memory accesses:

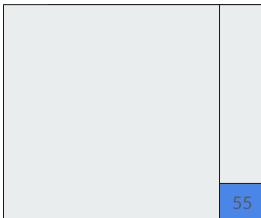
- Line buffer
- Window buffer

Data Stream with Custom Numeric Types

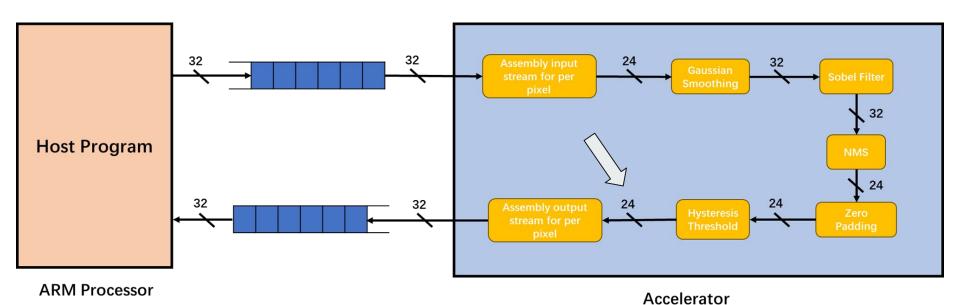




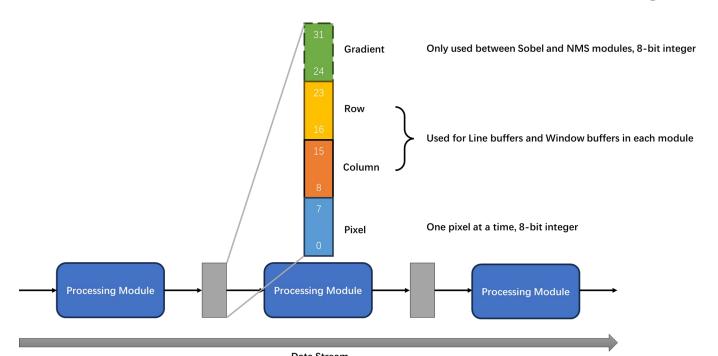




Stream with HLS Line/Window Buffer Design



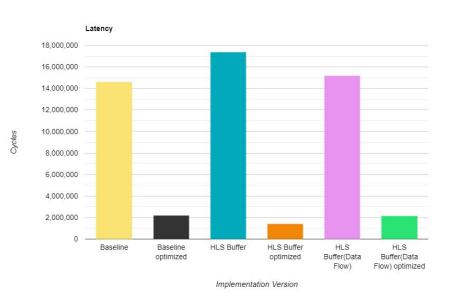
Stream with HLS Line/Window Buffer Design

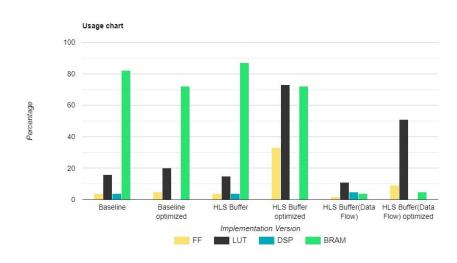


Resource Usage and Performance

	Latency(cycles)	FF	LUT	DSP	BRAM
Baseline	14599614	4	16	4	82
HLS Buffer	17370126	4	15	4	87
HLS Buffer(Data Flow)	15171585	2	11	5	4
Baseline optimized	2210652	5	20	0	72
HLS Buffer optimized	1391886	33	73	0	72
HLS Buffer(Data Flow) optimized	2146305	9	51	0	5

Resource Usage and Performance





Conclusion & Future Design

HLS acceleration

Line buffer & window buffer

Data Flow

Real-time Implementation