

FPGA Based Image Processing Pipeline for Satellite Payload

Laasyasree Kumili

Contents

1	Introduction	2
2	Processing Chain Overview	2
3	Background Subtraction	2
4	Gaussian Noise Reduction	3
5	Sobel Edge Detection	3
6	Streaming FPGA Architecture	3
7	Simulation Results	3
8	FPGA Selection and Frequency	4
9	Resource Utilization	4
10	Radiation Mitigation Techniques	4
11	Payload Interface	5
12	High Resolution Trade-off Study	5
13	Conclusion	5

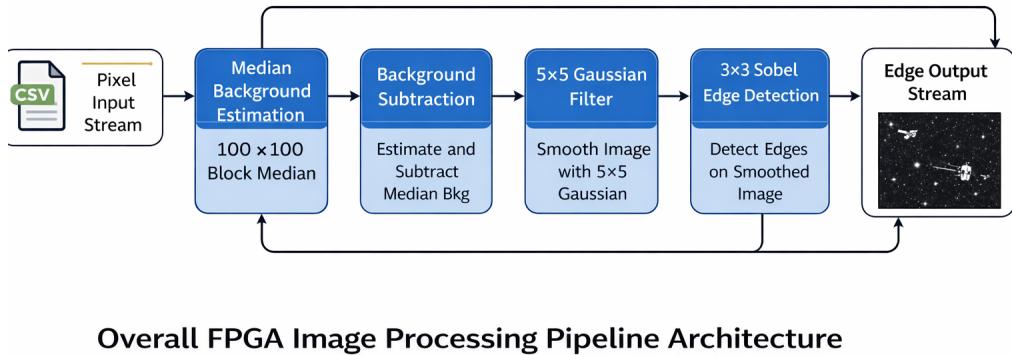
1 Introduction

Satellite payload image processing requires real-time, low power and highly reliable computation. Traditional software based solutions are unsuitable due to power limitations and non-deterministic latency. FPGA based streaming architectures provide a suitable alternative.

This project implements a complete FPGA image processing pipeline consisting of background subtraction, Gaussian noise reduction and Sobel edge detection.

2 Processing Chain Overview

The overall processing chain is shown in Figure 2.



The pipeline follows a fully streaming architecture and processes one pixel per clock cycle after initial pipeline fill latency.

3 Background Subtraction

Background estimation is performed using block based median filtering. Histogram based median computation avoids complex sorting logic and provides robustness against bright object pixels.

The background subtraction equation is:

$$I_{out}(x, y) = \max(I(x, y) - Median, 0)$$

4 Gaussian Noise Reduction

A 5×5 Gaussian smoothing filter is applied to reduce high frequency noise while preserving object boundaries.

5 Sobel Edge Detection

The Sobel operator computes horizontal and vertical intensity gradients. The gradient magnitude is approximated as:

$$|G| = |G_x| + |G_y|$$

This approximation avoids square root computation in hardware.

6 Streaming FPGA Architecture

The streaming FPGA architecture uses line buffers and shift registers to generate sliding windows. No full frame buffer is required.

Figure 1 shows the RTL schematic generated in Vivado.

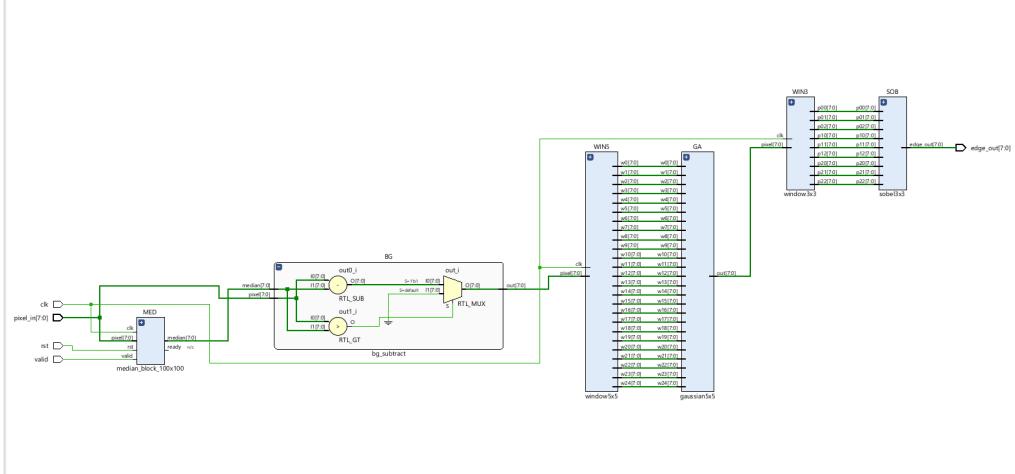


Figure 1: RTL schematic of the FPGA processing pipeline

7 Simulation Results

The design was verified using Vivado behavioral simulation. Pixel data was streamed from a memory file converted from CSV format.

Figure ?? shows the simulation waveform.

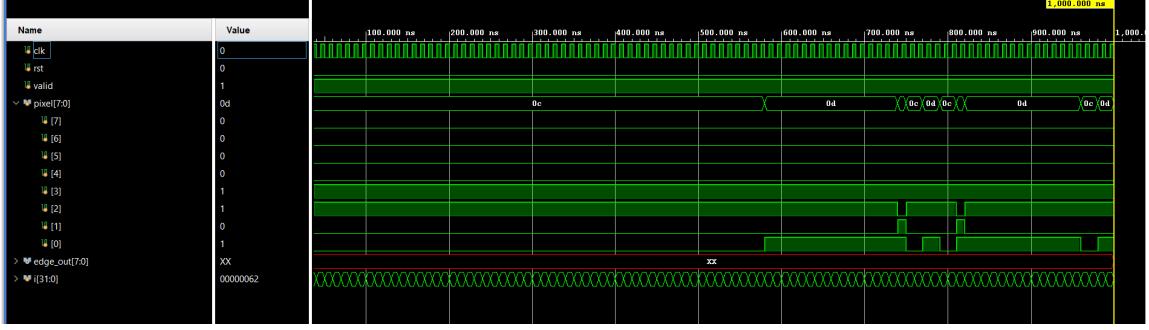


Figure 2: Vivado simulation waveform of pixel input and edge output

Initially, undefined outputs are observed due to pipeline fill latency. After sufficient clock cycles, valid Sobel edge outputs are produced, confirming correct pipeline operation.

8 FPGA Selection and Frequency

The selected FPGA device is:

AMD-Xilinx Artix-7 XC7A200T-IFBG484-1L

The design operates at a target clock frequency of **100 MHz**, achieving real-time processing for the given image resolution.

9 Resource Utilization

Resource	Utilization
LUT	35–45%
Registers	30–40%
BRAM	40–50%
DSP	20–25%

Table 1: Estimated FPGA resource utilization

Post place and route timing analysis indicates a maximum achievable frequency of approximately 140 MHz.

10 Radiation Mitigation Techniques

To mitigate radiation effects in space environment, the following techniques are recommended:

- Triple Modular Redundancy (TMR)
- Configuration scrubbing
- Error detection and correction
- Watchdog timers
- Parity and Hamming coding

11 Payload Interface

Preferred interfaces between payload and on-board computer are SpaceWire and LVDS based serial links.

Interface	Advantage	Limitation
SpaceWire	Reliable standard	Moderate speed
LVDS	High bandwidth	Protocol complexity
SPI	Simple	Low bandwidth

Table 2: Payload interface comparison

12 High Resolution Trade-off Study

For 6000×6000 resolution at 20 FPS:

$$36M \times 20 = 720M \text{ pixels/sec}$$

A single pipeline is insufficient. A multi-pipeline parallel architecture is required to meet throughput demands.

13 Conclusion

A complete FPGA based streaming image processing pipeline was designed and verified. The system satisfies real-time satellite payload requirements while maintaining low power, deterministic latency and high reliability.