Question 1 – RTL Module Development for Space Image Processing Pipeline

1.1 Overview

The satellite camera captures high-resolution deep-space imagery sized **3124 × 3030 pixels**, with each pixel represented using **8-bit grayscale**. The frame rate is **10 FPS**, meaning the processing system must sustain ~94.7 million pixels/second throughput.

Satellites must detect moving celestial objects (space debris, micro-satellites) which appear as **white streaks**, along with stars represented as **point sources**. Since background space is mostly black, the goal is to **remove static background noise and generate edge-enhanced frames** containing only relevant space objects.

This RTL pipeline must operate on-FPGA in real-time without software involvement.

Task-1: Background Subtraction RTL Design

Objective

Remove constant noise and illumination offset from the frame by subtracting a reference median intensity value computed over a **100×100 pixel sub-window**.

Design Approach

Step-1: Median Computation Block

- A rolling window buffer accumulates 10,000 pixels.
- A median extraction unit sorts values using histogram bucket accumulation.
- Pixel intensity range = 0–255 → histogram counter = 256 bins.
- The pixel intensity for which cumulative count > 5000 yields the **true median**.

RTL Expectations (background_subtract.sv)

- Input: pixel_in, pixel_valid
- Output: pixel_bg_removed = pixel_in - median

- Pipeline latency: 100–150 cycles depending on implementation
- Underflow handled using clamp(LUT): if result < 0 → set 0

This reduces noise as background illumination (space fog) is removed.

Task-2(A): Gaussian Noise Reduction (5×5 Kernel)

Purpose

Small pixel-level noise must be removed before edge extraction. A 5×5 Gaussian kernel is chosen for effective smoothing while keeping streak structure intact.

RTL Implementation Details

Kernel used:

1	**4**	**7**	**4**	**1**
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Normalization factor = **273**.

RTL Behavior (gaussian_filter.sv):

- 5-line buffer implemented using BRAM/shift-registers.
- Pixel convolution = Σ(kernel[i][j] × pixel[i][j])
- Division by 273 using DSP + shift approximation.
- Output: gaussian_out, gaussian_valid.

This operation smoothens stars for better Sobel response.

Task-2(B): Sobel Edge Detection (3×3 Kernel)

Kernel Used

Gx =

|−1|0|+1|

|−2|0|+2|

|−1|0|+1|

Gy =

|+1|+2|+1|

|0|0|0|

|−1|−2|−1|

RTL Dataflow (sobel.sv)

Magnitude = **abs(Gx) + abs(Gy)**

- Line buffer = 3 rows → 3×3 access per clock.
- Comparison thresholding sends only valid edges forward.
- Output: edge_val, edge_pixel_address.

Final Memory Output Stage

The pixel location and edge magnitude are written into **FPGA BRAM** for OBC retrieval.

```
IF edge_mag > threshold THEN
  BRAM[WRITE_ADDR] = {pixel_address, edge_value}
ENDIF
```

Frame storage capacity required:

Max edges/frame ~ 5–12% → approx ~5–10 MB worst case

Question 2 – Architecture, FPGA Selection & System Feasibility

Recommended FPGA Device

Device	Reason
Xilinx Kintex-7 / Artix-7 / XQR-Kintex radiation-tolerant	Balanced LUT/DSP count, low power, already space-qualified variants available
On-Chip BRAM + DSP48	Required for Gaussian+Sobel multiplies
Operating frequency @ 200-300 MHz	Far above minimum 100MHz needed for 10FPS processing

Space-Grade Design Considerations

- **TMR (Triple Modular Redundancy)** to prevent SEU corruption
- **Memory Scrubbing + ECC** for BRAM holding edge-frames
- **Watchdog fail-safe** resets if hang detected
- **Configuration scrubbing via ICAP**

This is critical for radiation and proton-flux exposure in LEO orbit.

On-board Output Interface Recommendation

Interface	Pros	Cons
SpaceWire	Industry standard for spacecraft, deterministic bandwidth	Hardware overhead
LVDS/SerDes	Low power, high throughput, simple	Requires OBC parallel capture
CAN/SPI	Light & easy	Too slow for full-frame output

Scalability Analysis (6000×6000 @ 20FPS)

Original Load = 3124×3030×10 = 94.7M pixels/sec

New Load = 6000×6000×20 = **720M pixels/sec ≈ 7.6× increase**

To Support Upgrade:

Upgrade Need	Strategy
7.6× throughput increase	Use multi-pipeline parallel Gaussian+Sobel cores
High memory bandwidth	DDR4/LPDDR + burst streaming
Thermal & Power Rise	Clock-gated sections + DVFS