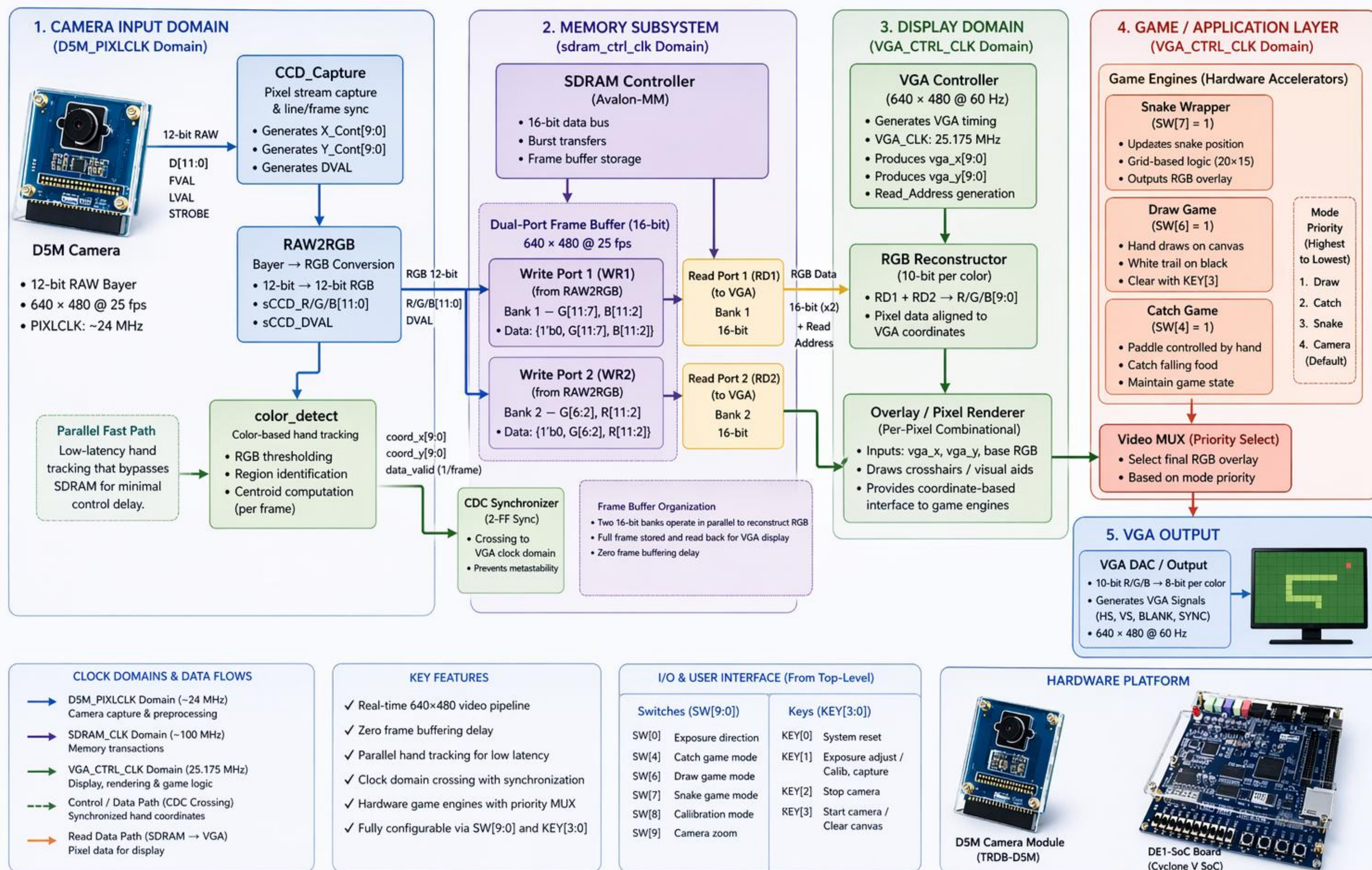


System Overview

System Architecture of Hand-Controlled Snake Game on FPGA

Real-time pipeline from D5M camera capture to VGA display with color-based hand tracking and game rendering

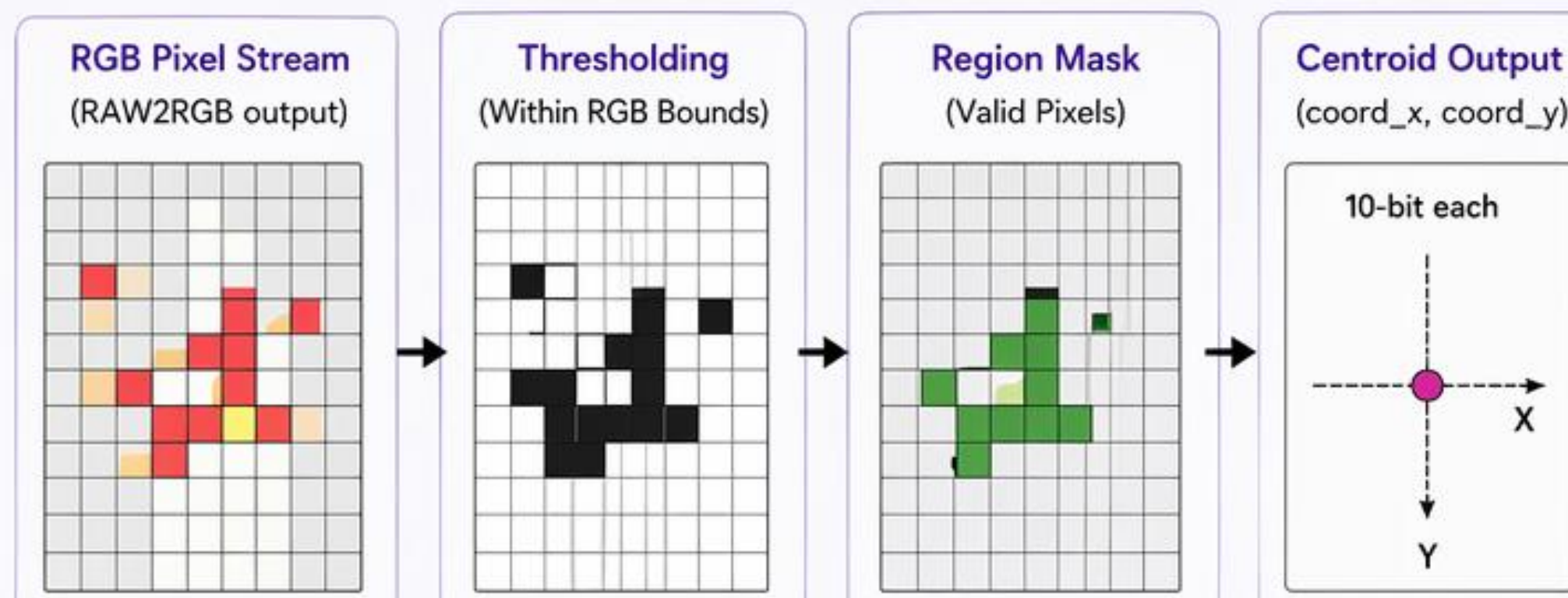


Core Algorithms

1. COLOR-BASED HAND DETECTION

Detects the user's hand in each frame using calibrated RGB thresholds and computes the centroid.

- RGB Thresholding**
Each incoming pixel (R, G, B) is compared to calibrated min/max bounds.
- Region Mask Generation**
A binary mask is produced where pixels within the bounds are marked valid.
- Accumulate & Count**
For every valid pixel at position (x, y):
sum_x += x
sum_y += y
count += 1
- Centroid Computation (per frame)**
When the frame is complete (FVAL falling edge):
centroid_x = sum_x / count
centroid_y = sum_y / count
If count = 0, centroid outputs previous value.



Calibration Mode (SW[8] = 1)

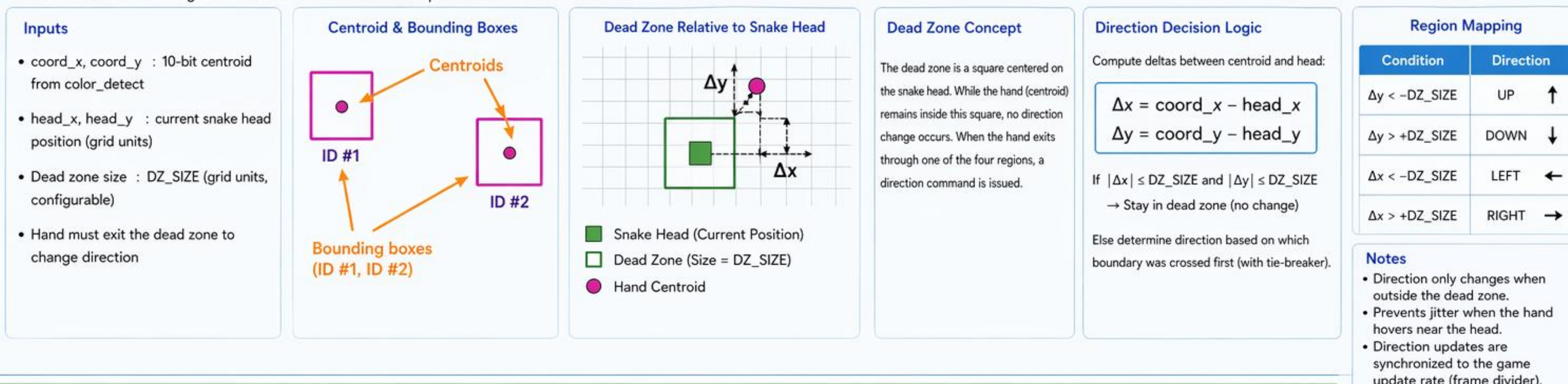
- The center 32×32 block is averaged to obtain (R, G, B) values.
- KEY[1] captures these values as new thresholds.
- HEX displays the averaged RGB during calibration.

Output Update Rate

Centroid updates once per frame (25 fps) and is synchronized to the VGA clock domain before being used by the control logic.

2. CENTROID OUTPUT & DEAD ZONE (SNAKE HEAD)

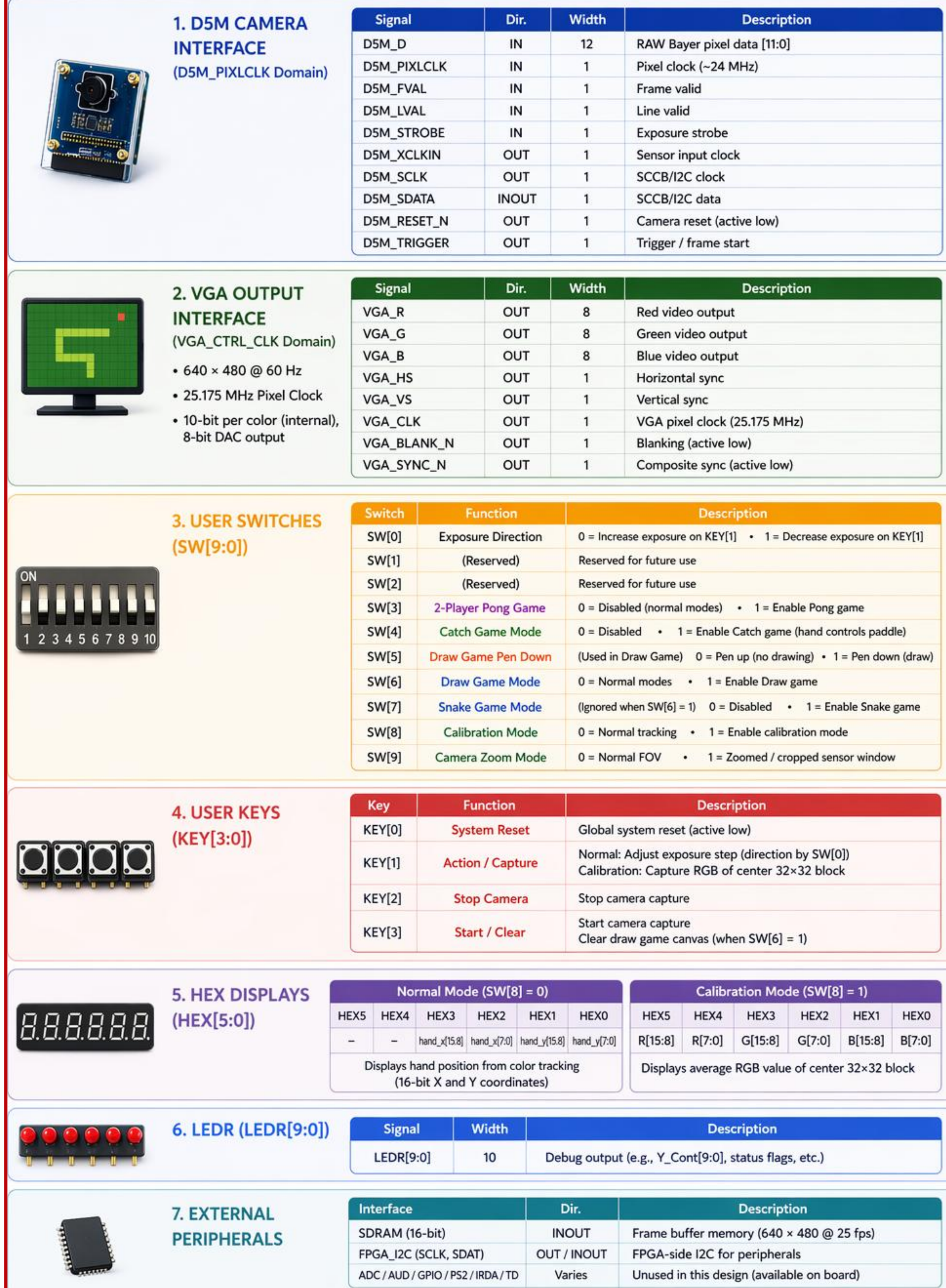
The centroid represents the hand position. The snake head defines a moving dead zone. Direction commands are generated based on the centroid's position relative to the head.



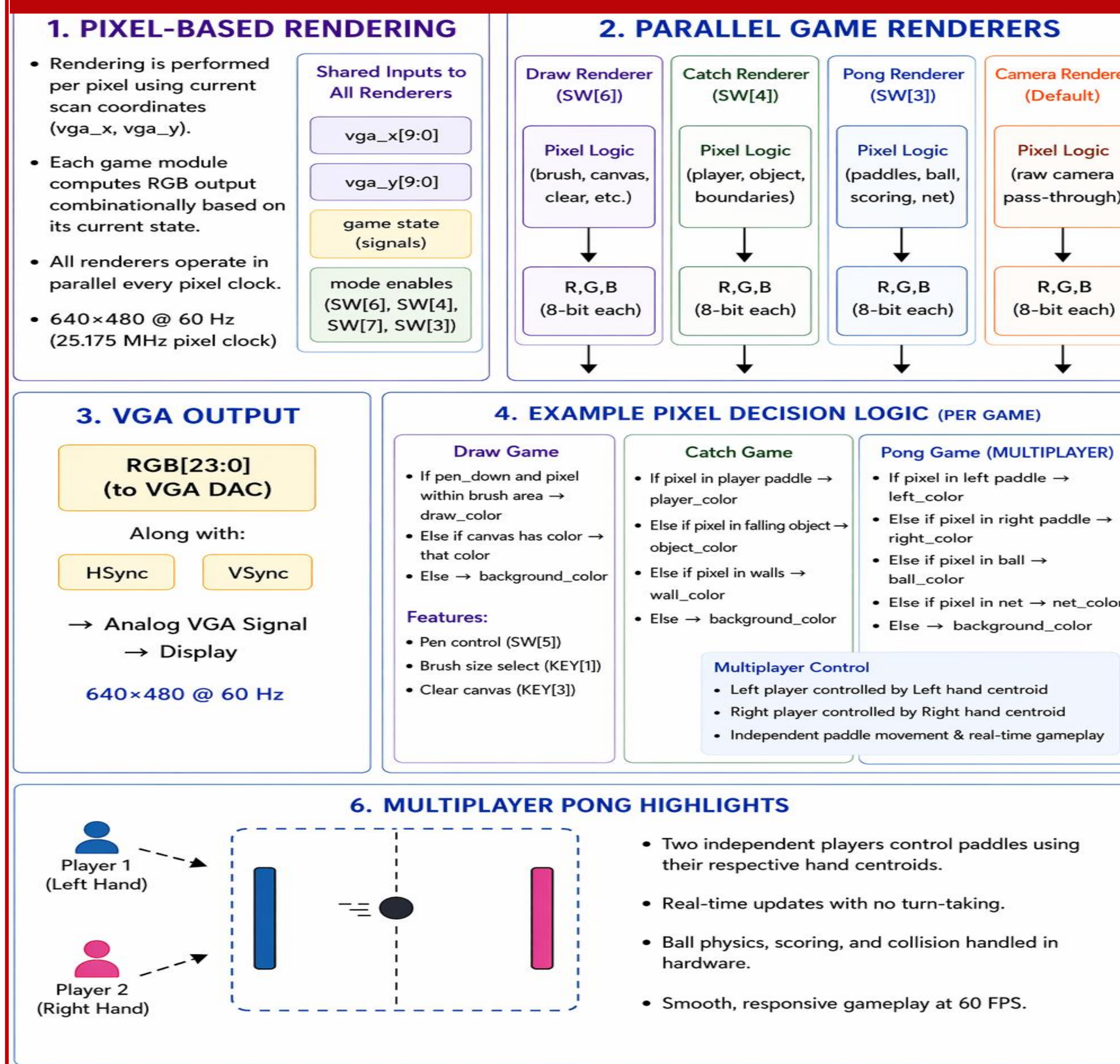
Challenges

- Clock Domain Crossing (Camera → VGA):** Safe synchronization was required to transfer hand tracking data between different clock domains without instability.
- Latency vs. Responsiveness Tradeoff:** Frame-based centroid updates improved stability but introduced slight delay in control response.
- Designing an Intuitive Control Scheme:** A moving dead zone around the snake head was needed to reduce jitter and enable natural gameplay.

Hardware Interfaces & Peripherals



Rendering Pipeline & Game Output



Future Considerations

- AI/ML-Based Hand Tracking:** Replacing RGB thresholding with ML techniques could improve robustness to lighting changes and enable gesture recognition.
- Enhanced Hardware Platform:** Upgrading to a more powerful FPGA or camera would allow for better performance and graphics
- Expanded Gameplay:** Adding more advanced game mechanics, scoring systems, AI opponents, and modes would enhance overall user experience.