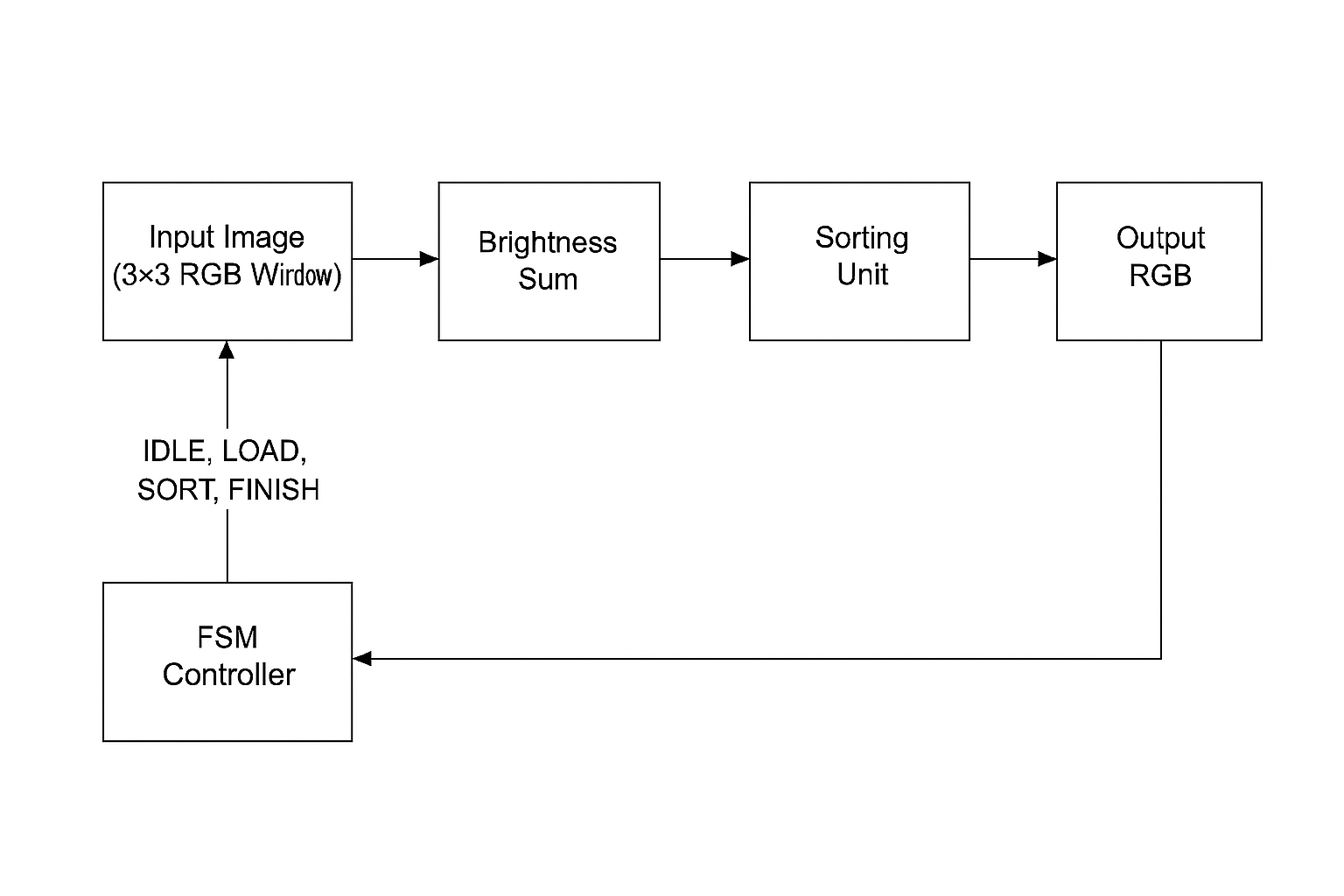
The Block diagram provides an overview of the Verilog design’s architecture



A. Input Image (3×3 RGB Window) :

* In this block, a 3x3 block of pixels in being input. This 3×3 matrix represents the pixel neighbourhood around a central pixel in the image. Each pixel has 8-bit R, G, and B channels, stored in registers for processing.

B. Brightness Sum

* Computes R + G + B for each of the 9 pixels in the window to calculate the brightness of each. This produces 9 brightness values (brightness [0] to brightness [8]). One for each pixel that determines the median pixel in terms of brightness. Each brightness is 24 bits, stored in temporary registers for sorting later.

C. Sorting Unit

* Implements bubble sort on the 9 brightness values, arranging them from lowest to highest. Brightness values get swapped, and their R, G, and B values also get swapped in parallel, so they stay aligned with their brightness. The sort runs over multiple clocks, comparing and swapping until it is complete, then the median pixel will be at index 4 of the sorted arrays.

D. Median Extraction (Output RGB)

* After sorting, the pixel at index 4 is selected as the median. This is then output as the new value of the centre pixel.

E. FSM Controller

* Manages the operation across 4 states:
  + IDLE: Waits for start signal, then initializes state and outputs unmodified 3x3 input window.
  + LOAD: captures 3x3 RGB inputs into window registers and computes brightness sums for each of the 9 pixels.
  + SORT: Performs bubble sort across the brightness array.
  + FINISH: Writes the median and asserts done from sorted arrays into the output registers.