

Image Browser



Mary Knapp
Prannay Budhreja

Overview

- To build an Image Browser:
 1. VGA Module
 2. USB Reader
 3. Processing Module
 4. Memories
 5. GUI
 6. Finite State Machine
 - Add switches and pipelines to taste

Memory and Resolution

- Two frame buffers
- Required memory = $2 * (\text{height} * \text{width}) * 24$
- 1024x768? Requires 4.7Mb → ☹️
- 800x600? Requires 2.88Mb → 😊
- Choose SVGA

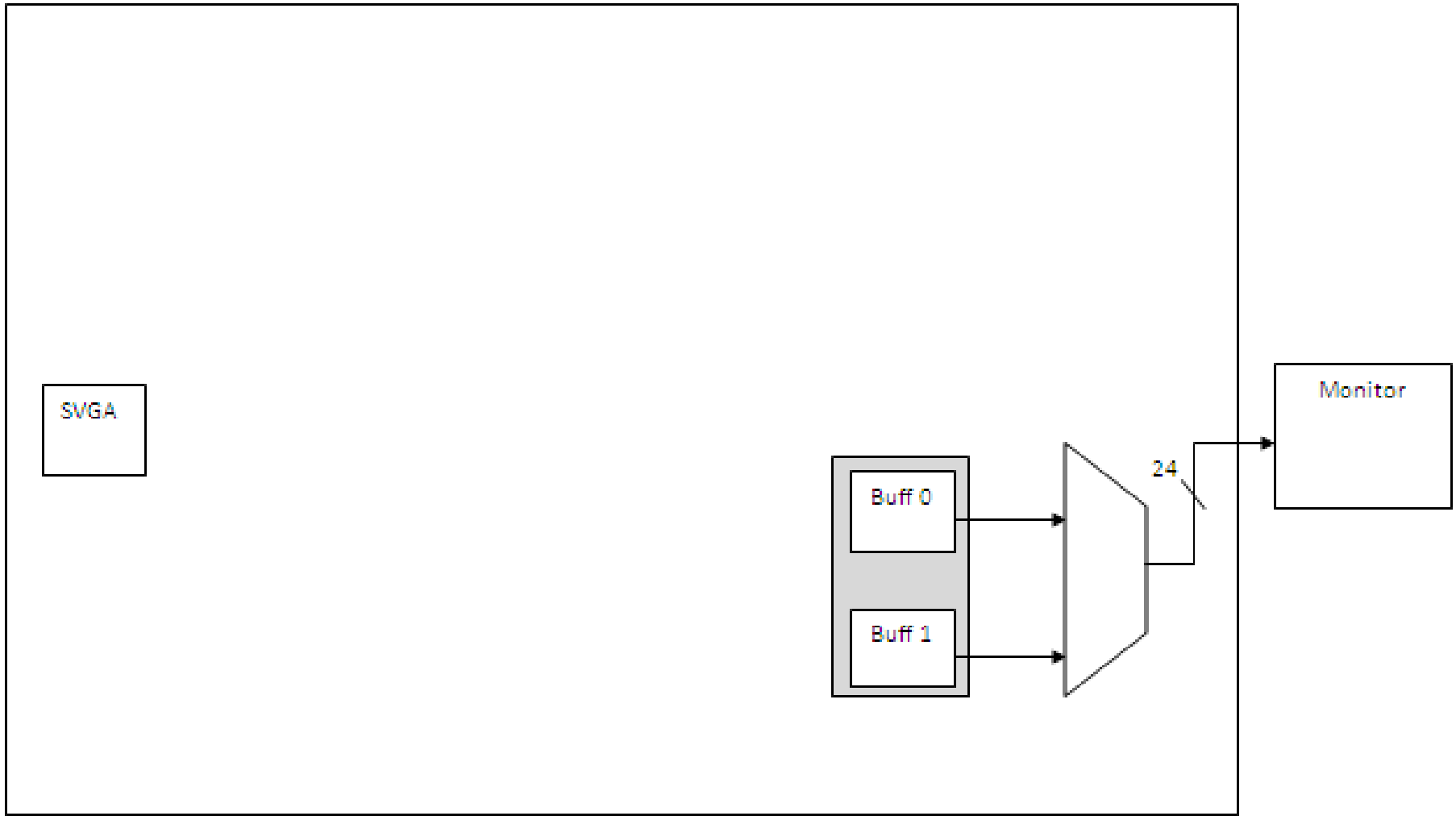
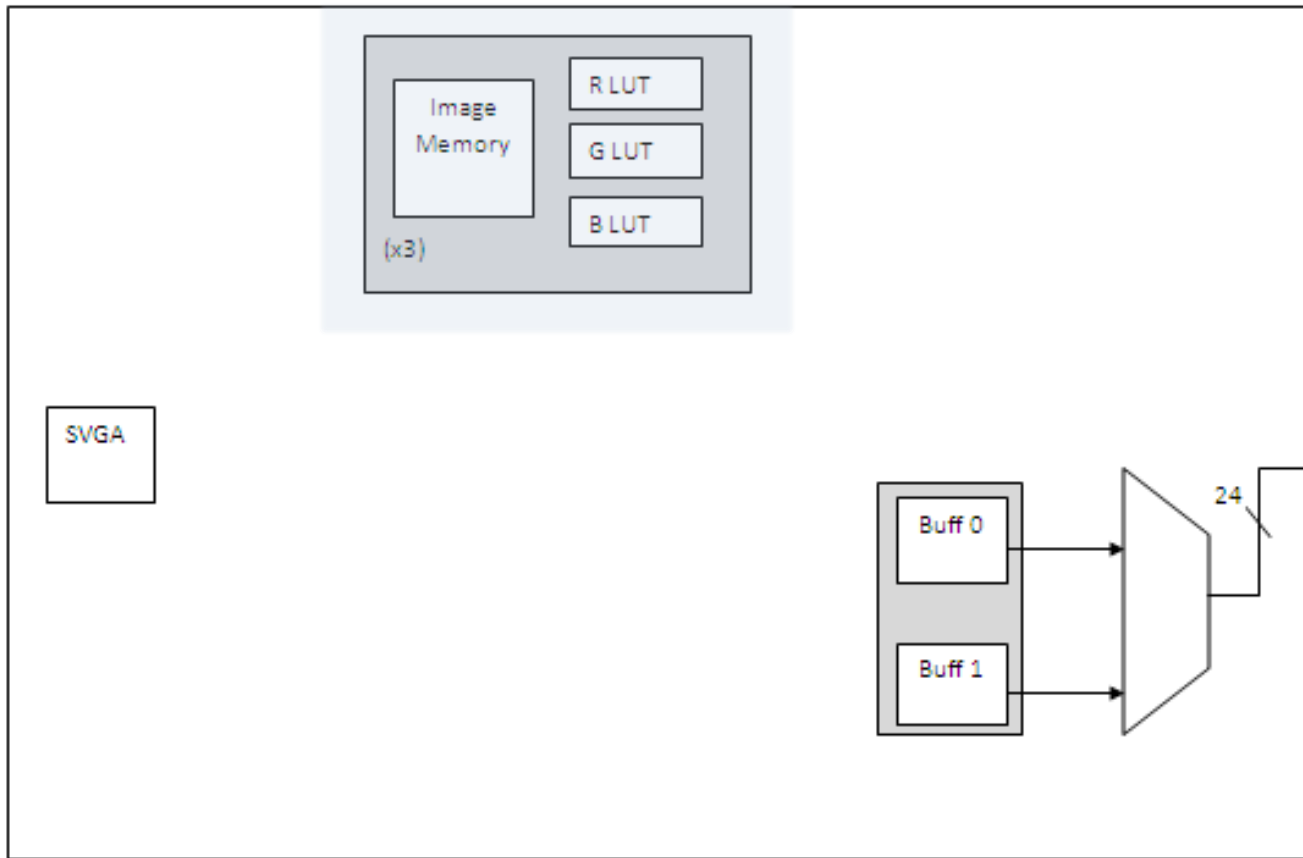
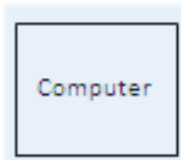


Image Size and Number

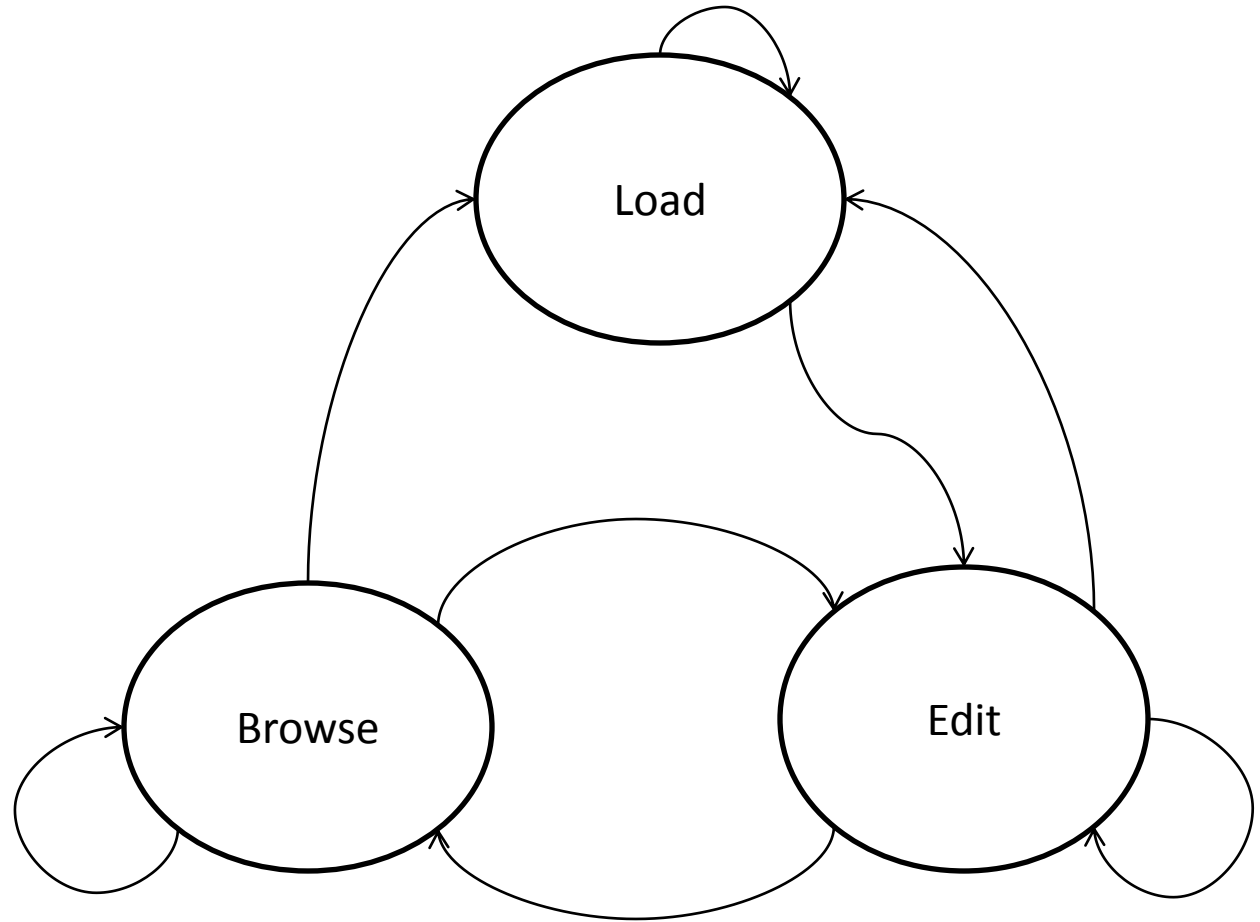
- Compress images → 8-bit image (256 color) + color map (MATLAB)
- Three LUT per image (RGB)
- Size limit: 640x480
- Can store 3 images

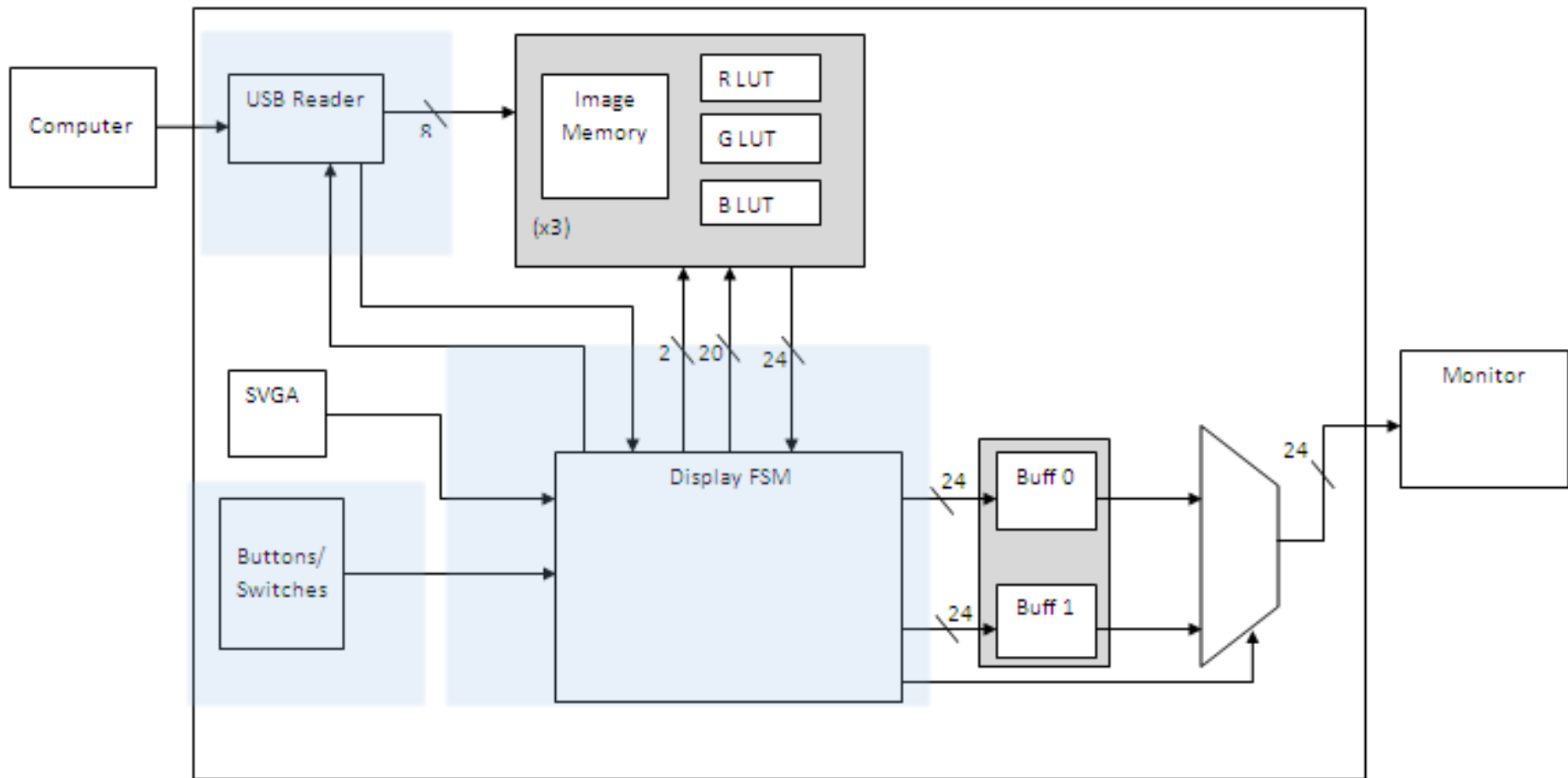


Display FSM

Three states:

- Load
- Browse
- Edit



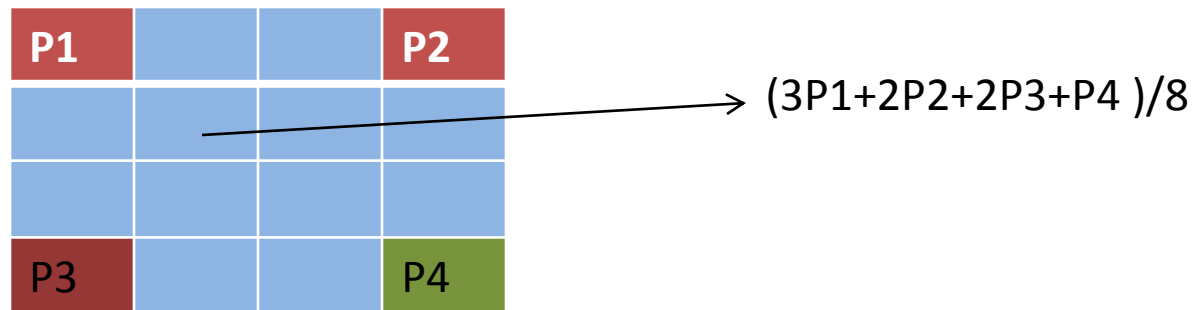


Transformations

- Scaling
 - Anti-Aliasing: Zooming out. Displaying a high-res image at a low resolution.
 - Interpolation: Zooming in. Displaying more pixels than image resolution.
- Rotation
 - Nearest Neighbor Rotation
 - Rotation by Sub-pixel Area Average

Scaling

- Anti-Aliasing
 - Bi-Linear → considers pixels on all 4 sides
 - Coefficients in ROM → only certain levels of zoom out allowed
 - Without optimization it can take $O(pS^2)$. p = #output pix, S = scale factor
- Interpolation
 - Bilinear → uses a 2x2 pixel block

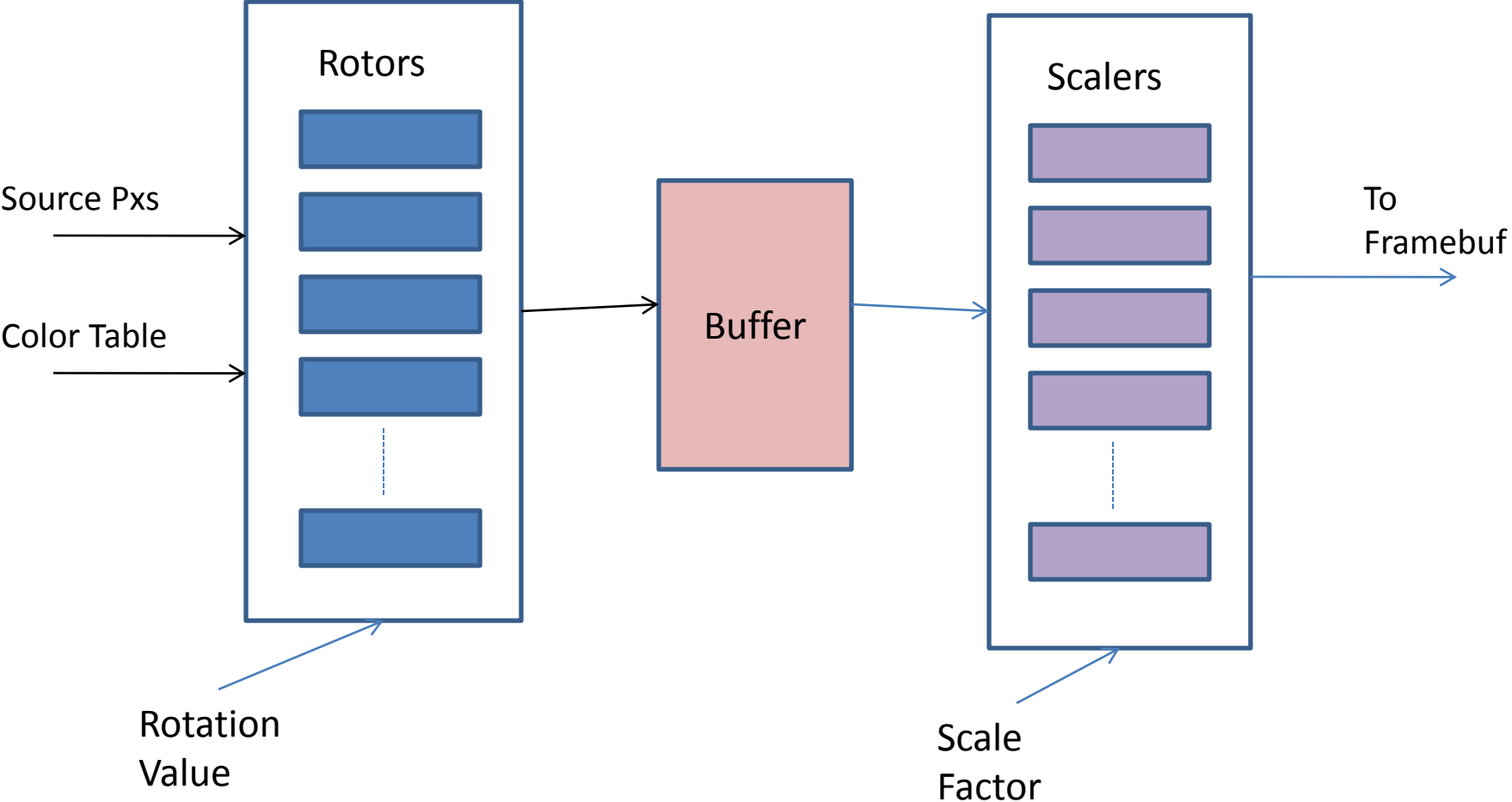


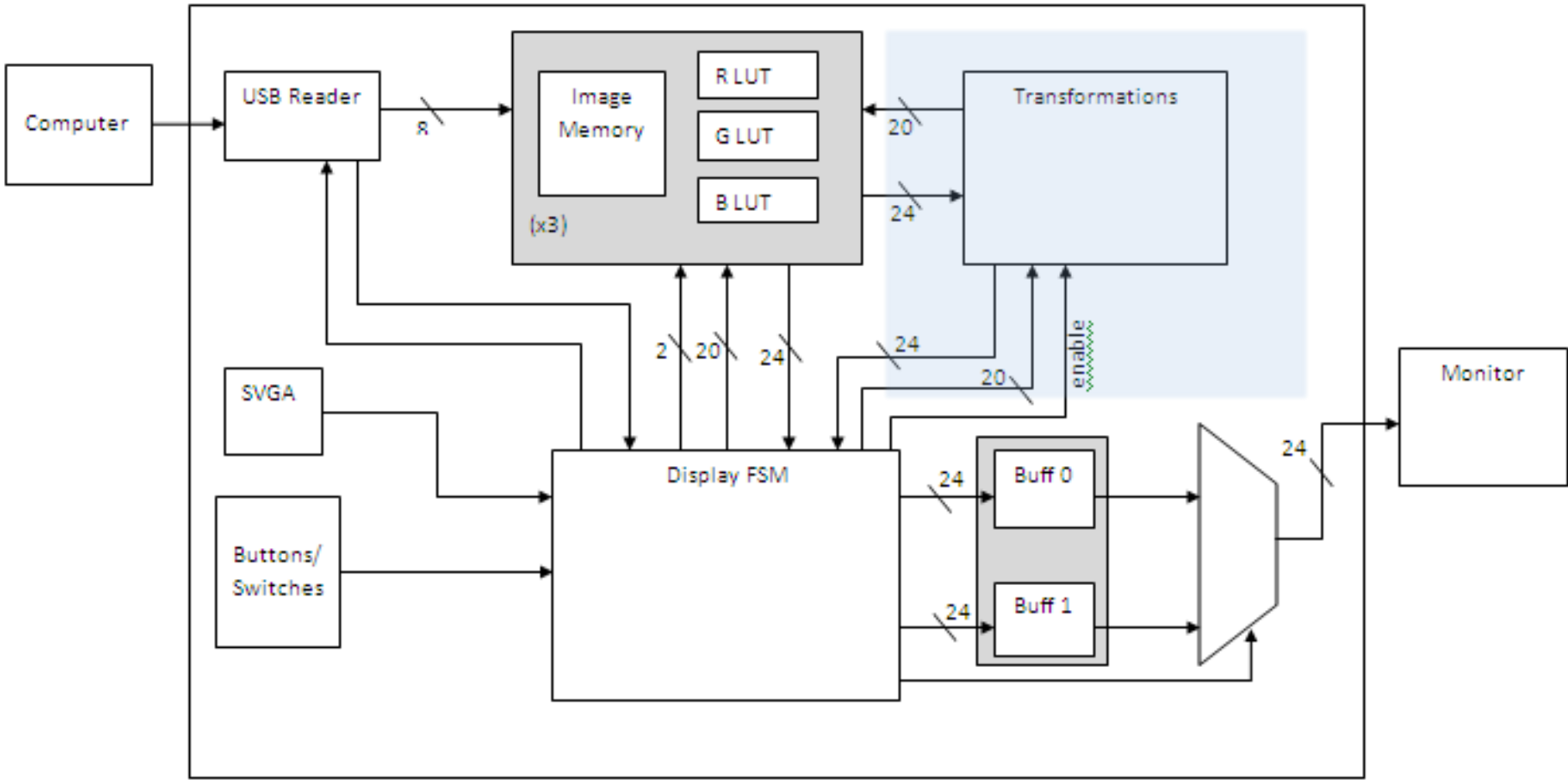
Timing Constraints

- Interpolating a 2x2 pixelblock requires (S ==scale factor):
 - S^2 coefficients
 - 4 Multiplications/output pixel

=> $4S^2$ signed decimal multiplications / block
- 1/60 sec Frame rate with system clock @65Mhz → 10^6 clock cycles
- 640x480 = 300k pixels → 3 cycles/pixel ☹️
- Solution: Pipelining

Block Diagram





Timeline

Week	Mary's Goals	Prannay's Goals
Nov 16	Load+Display Images	Software Complete + Verilog Prototype
Nov 23	USB Reader + GUIs + Switching	Verilog + Testing
Nov 30	Finished GUI + Scrolling	Debugging
Dec 7	Debugging	Gesture Recog