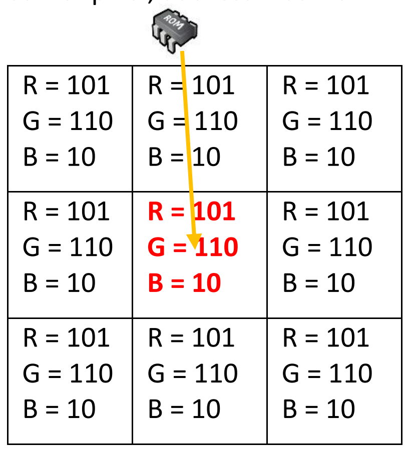
Project structure

Initially, we used nine different onboard ROM to store one particular picture nine different times. Then we pull out color values of eight pixels which surround center by using different address instruction to the rest of eight different memories. Next step user will give the kernel value for a 3X3 kernel matrix. We take those values by our finite state machine and then doing 2-D convolution before output color value to VGA. Final step is output the color of center pixel after finish computation

center pixel, address = center



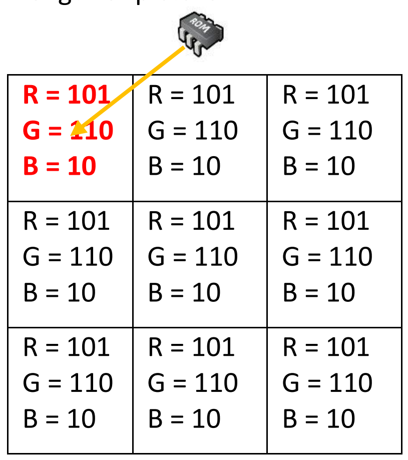
R = 101
G = 110
B = 10
	R = 101
	G = 110
	B = 10
		R = 101
		G = 110
		B = 10

R = 101
G = 110
B = 10
	**R = 101
	G = 110
	B = 10**
		R = 101
		G = 110
		B = 10

R = 101
G = 110
B = 10
	R = 101
	G = 110
	B = 10
		R = 101
		G = 110
		B = 10

surround pixel-1, address1 = center

- length of picture - 1



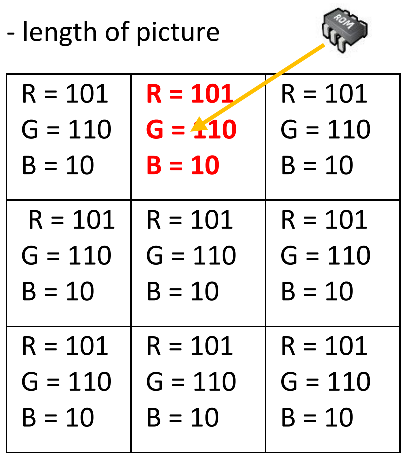
**R = 101
G = 110
B = 10**
	R = 101
	G = 110
	B = 10
		R = 101
		G = 110
		B = 10

R = 101
G = 110
B = 10
	R = 101
	G = 110
	B = 10
		R = 101
		G = 110
		B = 10

R = 101
G = 110
B = 10
	R = 101
	G = 110
	B = 10
		R = 101
		G = 110
		B = 10

surround pixel-2, address2 = center

- length of picture



R = 101
G = 110
B = 10
	**R = 101
	G = 110
	B = 10**
		R = 101
		G = 110
		B = 10

R = 101
G = 110
B = 10
	R = 101
	G = 110
	B = 10
		R = 101
		G = 110
		B = 10

R = 101
G = 110
B = 10
	R = 101
	G = 110
	B = 10
		R = 101
		G = 110
		B = 10

6 more

kernel

0.0625	0.125	0.0625
0.125	0.25	0.125
0.0625	0.125	0.0625

Output color value of center pixel = cov(all surround pixels and center itself with kernel)