Hardware Implementation of a 1D and 2D Convolver

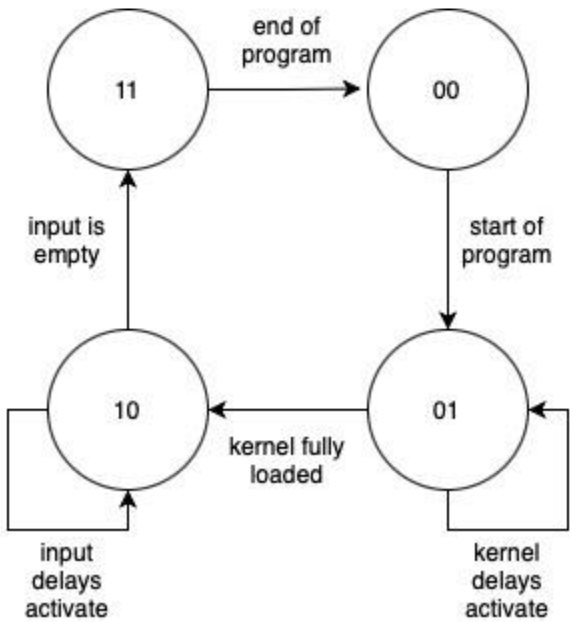
Introduction:

For this project, we were tasked with designing the hardware for a one-dimensional convolver, and then extending that to two-dimensions. The first step of this process was understanding what a convolver's function is. Convolution is a mathematical operation which can be used for various operations, such as filtering, edge detection, and feature extraction. Convolution uses a small matrix, referred to as the kernel, and applies it over an image matrix. The kernel is first flipped, and then slid over the image matrix. At each position of the image matrix, each value of the kernel is multiplied to the corresponding value on the image, and then they are all added together, resulting in one value. That value is then assigned to the image matrix where the center point of the kernel was positioned. The kernel then continues to slide across the image matrix until it has gone across each value, resulting in a new output image.

1D:

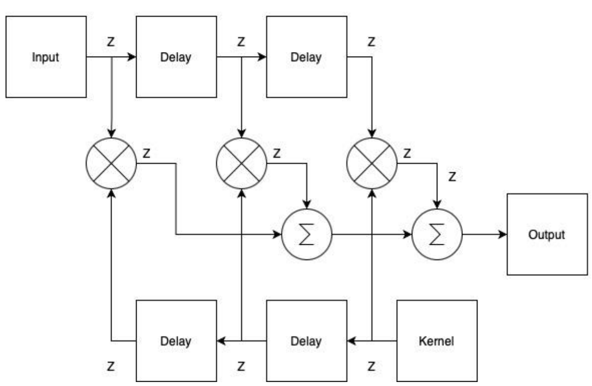
Implementing this process in hardware in one-dimension, we first needed to flip the kernel. To flip the kernel in hardware, the kernel vector can be fed through the hardware using shift registers in the opposite direction of the image vector. The kernel goes through N shift registers, where N is the size of the kernel. Similarly, the input image would be fed through N shift registers, but in the opposite direction. We chose to base the length of the convolver on the length of the kernel because we had direct control over its size. That is not the case for the input image. After each of the shift registers for both the kernel and image, there would be a multiplier. The segments after the shift registers of the kernel and image are both connected to these multipliers. This allowed for the multiplication of the kernel and corresponding image value. Lastly, the output was the result of the summed products from each of the multiplication blocks.

State Machine:

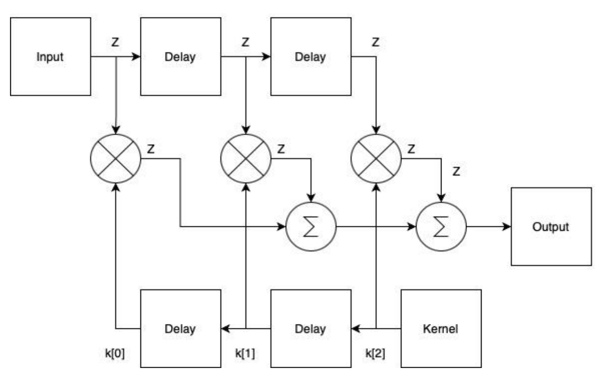


Example With Steps:

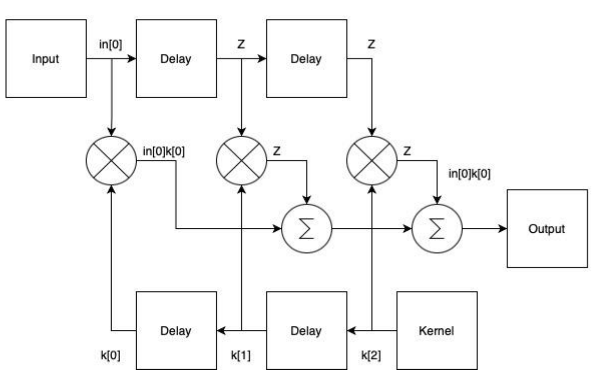
Step 0



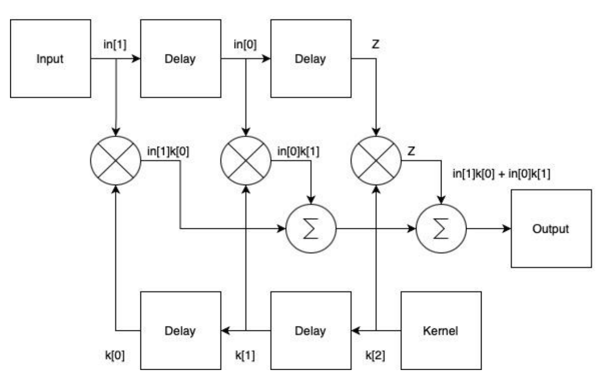
Step 1



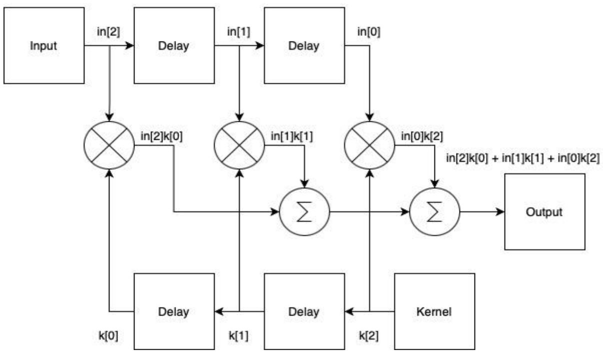
Step 2



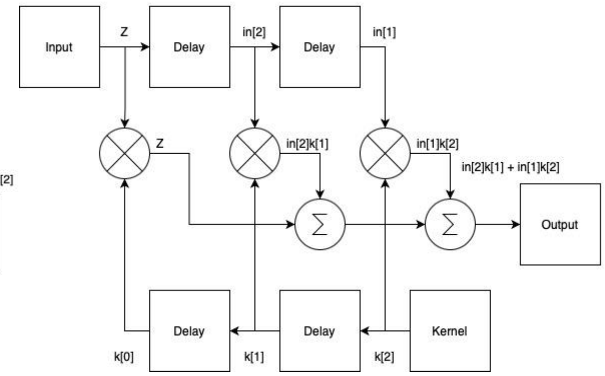
Step 3



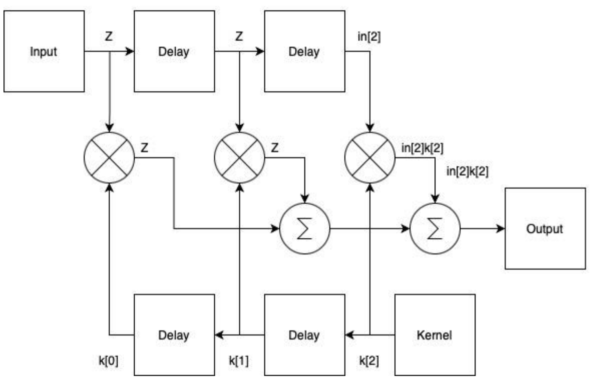
Step 4



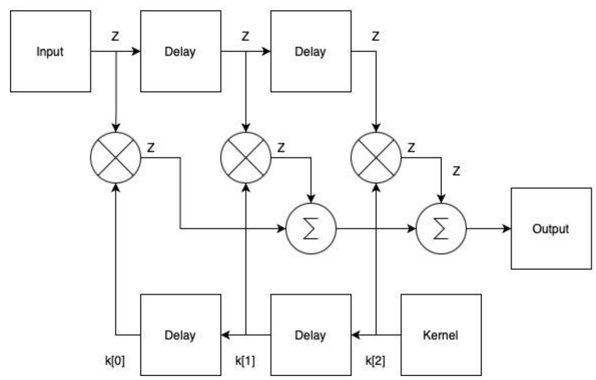
Step 5



Step 5



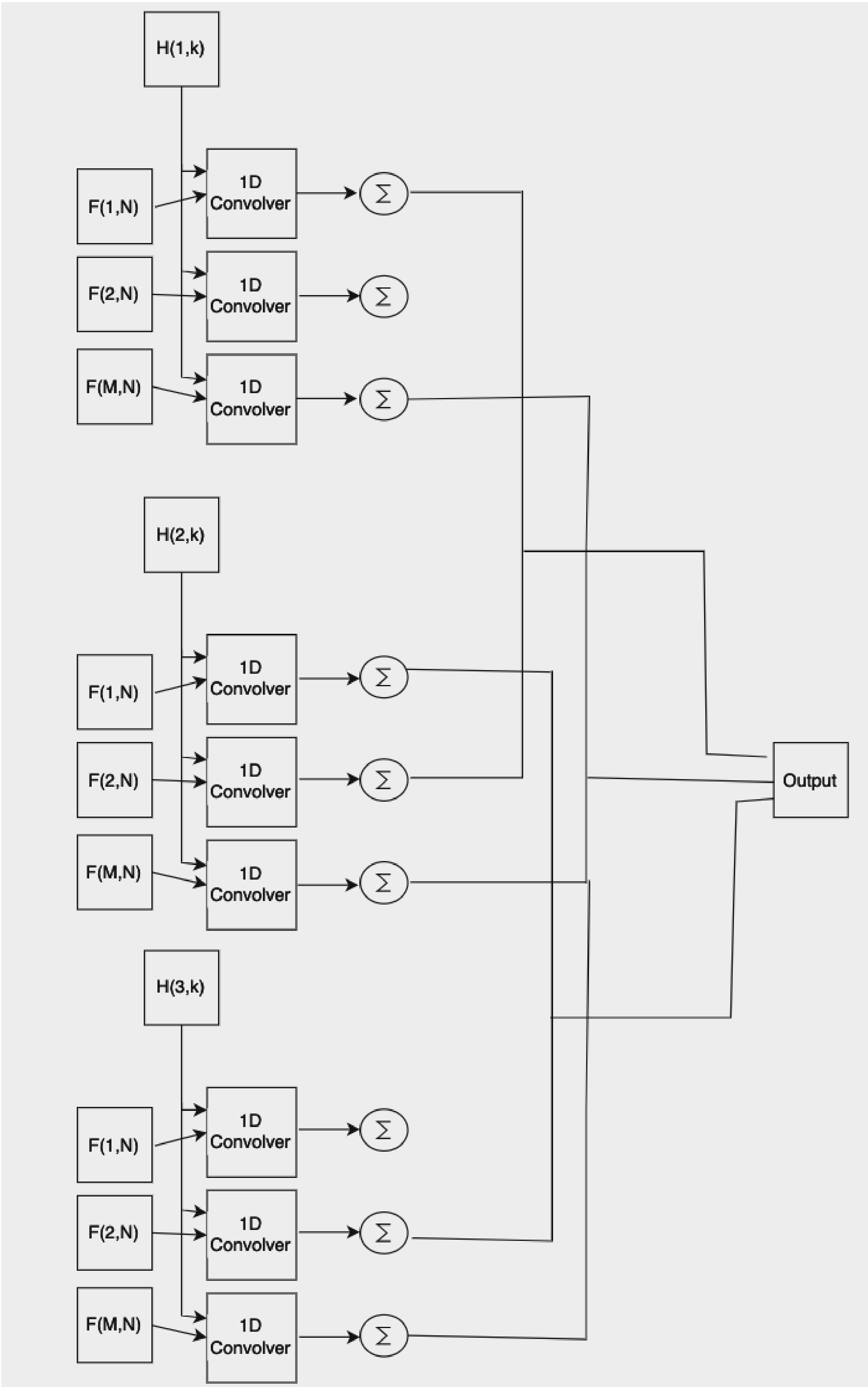
Step 7. Convolution Done



2D:

Reflecting this process into two dimensions, the kernel and image would then both be matrices. With these new added rows, our hardware would need to have more of the one-dimensional convolvers. To define a 3x3 kernel, there needs to be 3*M one-dimensional convolvers, where M is the number of rows of the image matrix. This is to allow for an iteration of each row of the kernel to convolve with each row of the image matrix. Next, there needs to be an adder for each shift register on the outputs of each one-dimensional convolver. Then, each of the adders would be connected for each of the kernels. Better put, the adder of the first-row kernel with the first row of the image matrix would be connected to the second-row kernel with the second row of the image matrix, and the same with the third-row kernel. Adding these together would give one number, and eventually put into a matrix, resulting in a new image matrix.

Diagram:


H(1,k)
F(1,N)
1D Convolver
Σ
F(2,N)
1D Convolver
Σ
F(M,N)
1D Convolver
Σ
H(2,k)
F(1,N)
1D Convolver
Σ
F(2,N)
1D Convolver
Σ
F(M,N)
1D Convolver
Σ
Output
H(3,k)
F(1,N)
1D Convolver
Σ
F(2,N)
1D Convolver
Σ
F(M,N)
1D Convolver
Σ