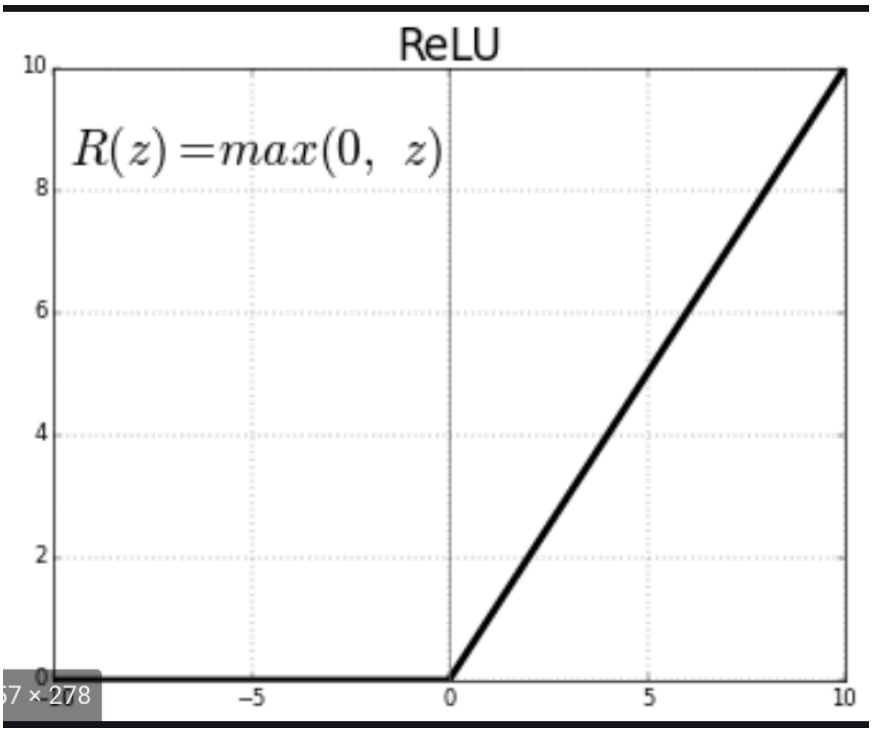
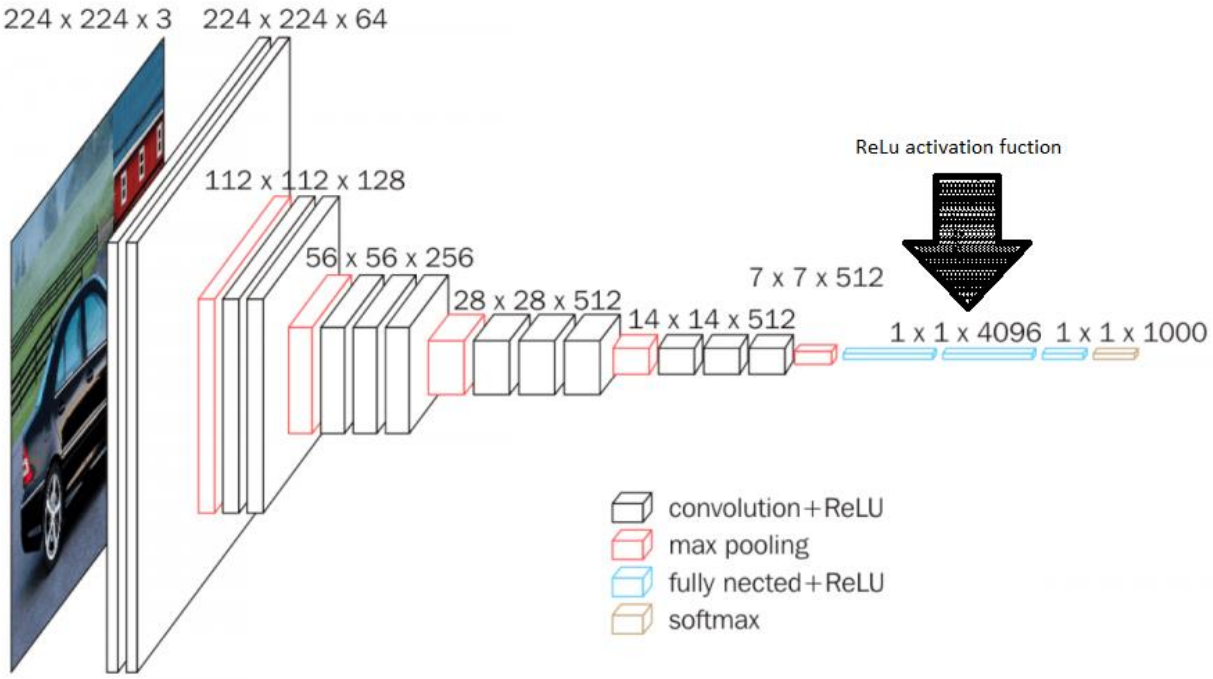
ReLU Activation function



ReLU block diagram



Input of the block is the stream of bits all as 1 array without grouping and the output is the same but after rectifying the image

Code Idea

The output(max_temp) is the same as the input(d_in) if the input is greater than 0

But it will be zero if in the negative area

```
generate   //counter such that for each array to begin rectifying it
        for (i=0; i<Rows; i=i+1) begin : out_num
                always @ (posedge clk)
                        if (!rstn)                              max_temp[i] <= 0;
                        else if (d_in[i][MSB])                  max_temp[i] <= 0;
                        else                                    max_temp[i] <= d_in[i];
        end
endgenerate
```

For each array if the most significant bit =1 (negative number)as we are using signed bits, this array will be equal to 0 (maxtemp<=0)

The image is divided into stream of bits each stream of bit is described as array and looping around every array to determine whether the MSB is 1 or not

Example:

100101

010001

001001

100101

So this is a 4 stream of bits(Rows variable) with 6 bits each (Columns variable)

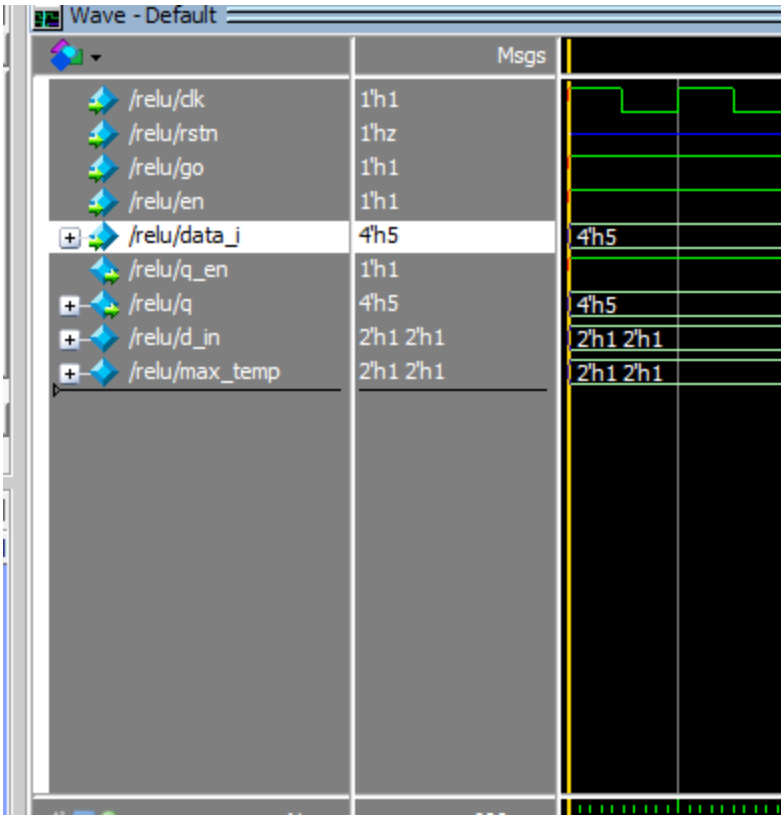
Simulation Results

Here on a small scale to simplifiy simulation we divided it into only 2 arrays (2 bits and 2 bits)

Input:0101

Expected output: 0101

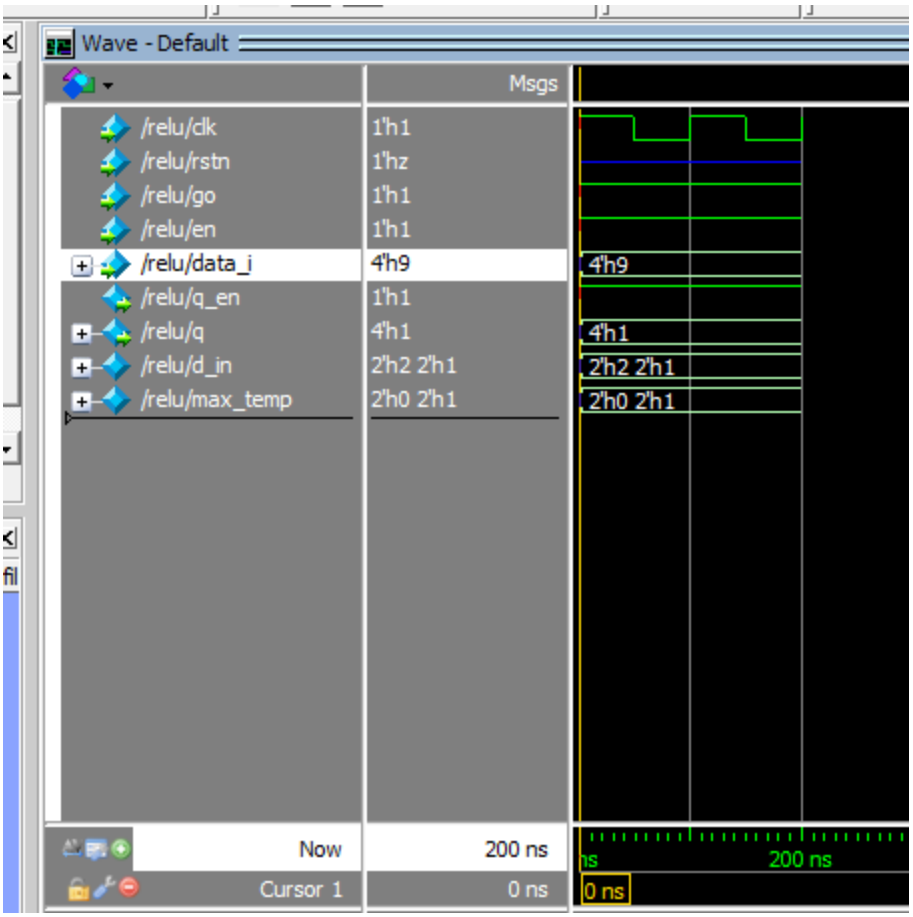
In hexa: 5



Input: 1001

Expected output: 0001

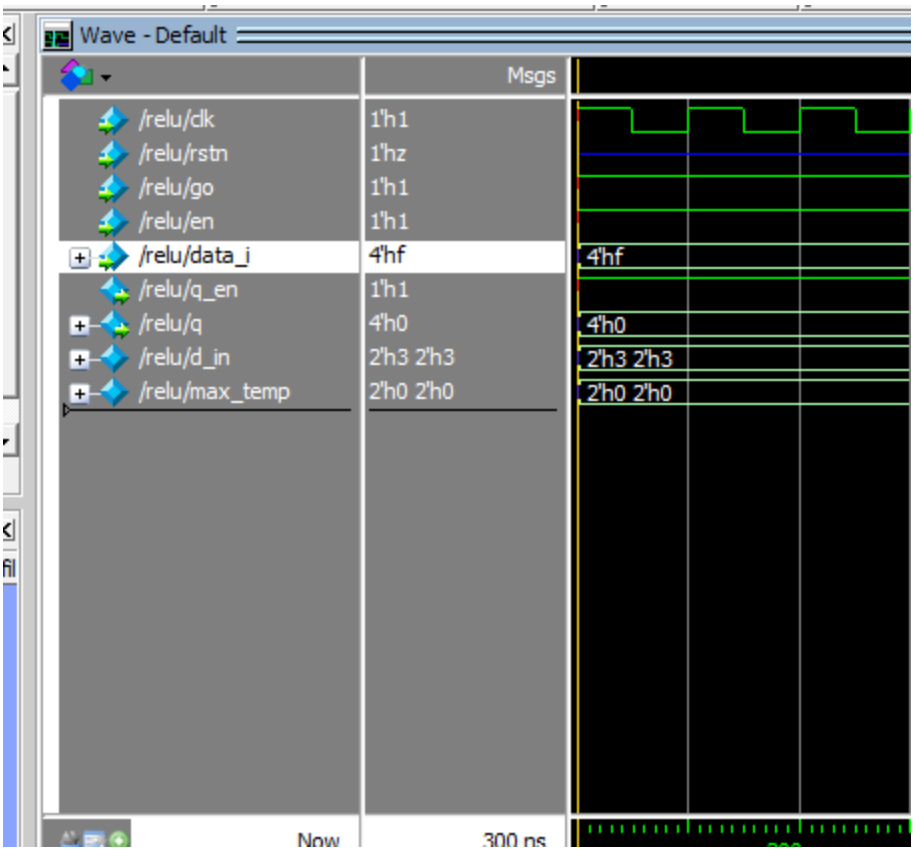
In Hexa: 1



Input: 1111

Expected output: 0000

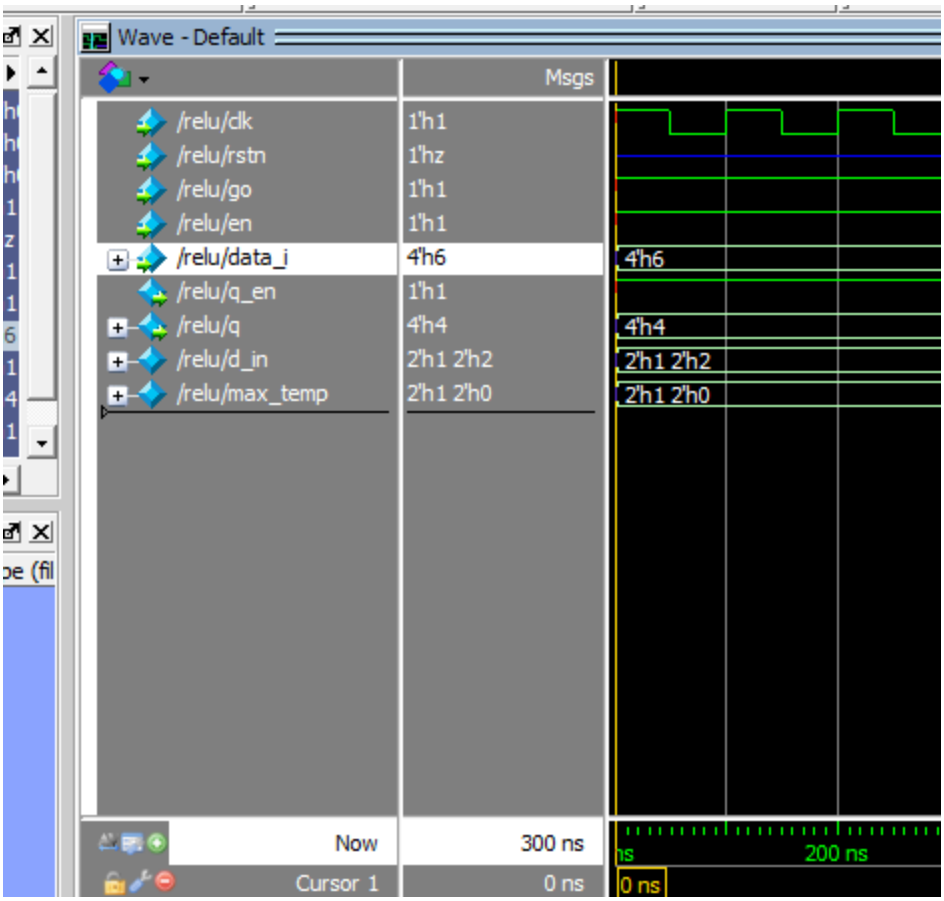
In Hexa: 0



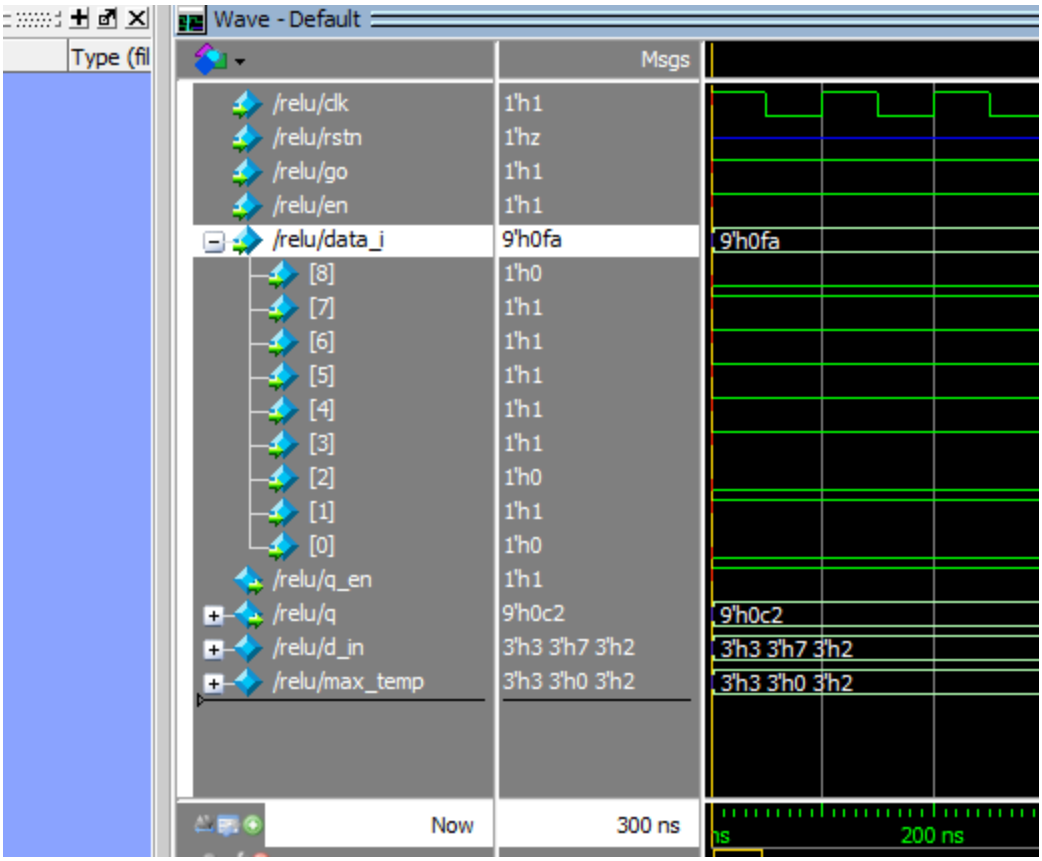
Input: 0110

Expected output: 0100

In Hexa: 4



Taking on a bigger scale with 3 stream of bits with 3 bits each



The input is “011 111 010” which is in hexadecimal “FA”.

So the expected output is ”011 000 010”

Which is in hexadecimal “C2”