COL215 - Hardware Assignment 3 (Part-1) Report

Abhishek Amrendra Kumar 2022CS11598
Ayush Kumar Gupta 2022CS11114

2023-11-08
Group No-2
Submission Time- 08:00 AM

As our lab is on Tuesday, Ms Radhika Mam had extended deadline for Group 2 till 11:59 AM of 8 Nov

1 Aim of the Assignment

The objective of the assignment is implementation of a 3x3 image filtering operation, which is an extension of the previous hardware assignment. The components involved are memory elements (RAM, ROM and registers), compute unit (filter operation), and VGA controller.

2 Filtering Operation

2.1 Algorithm Explanation

In the previous assignment, we have already done the gradient operation. This assignment was also somehow related to that. But in this assignment the actual challenge was to create MAC units and using it for filtering with 3x3 kernel.

For tackling this challenge, we first made a compute unit which comprises of 9 MAC units(We have explained MAC units in next section). Since matrix multiplication is to be performed, it contains 9 multiplications and 8 additions. For Each multiplication, we have made a MAC Unit, which computes the product of the 8-bit signed value of the filter and the 8-bit unsigned value of the pixel. Lastly, they all are added together to get the final result(which is efficiently done by the Compute Unit).

The Final module is instrumental in carrying out the gradient operation. Its core function revolves around acquiring data from the filter, structuring it within a memory vector to facilitate swift access, and methodically traversing a 64x64 grid. At each iteration, all nine data points within the grid are extracted, with meticulous attention given to scenarios involving edge cases. Although the use of the MAC (Multiplier-Accumulator) unit might not be immediately apparent within this module, its significance becomes pronounced when we delve into the realm of the Finite State Machine (FSM) segment.

In the realm of clock cycle analysis, the procedure unfolds as follows: Initially, all values are requisitioned from the Filter. This operation necessitates approximately 18 clock cycles to delineate the addresses from which the values are to be retrieved. An additional series of iterations is required to retrieve these values physically. Subsequently, for each instance of fram array, a total of 3 clock cycles is invested in comprehensively computing per pixel, all the while perpetually accumulating it. This culminates in the data being inscribed into RAM within the last cycle. Consequently, when contemplating the overall number of clock cycles for each iteration, the average is computed as (3 + 18/4096).

3 Explanation of Compute Unit and MAC Unit

The Multiplier-Accumulator (MAC) unit is a critical component of our image-filtering system. It plays a central role in the image processing operation, where it takes input values, multiplies them, and produces a single output.

The MAC unit comprises the following key components:

1. **Multiplier**: This component is responsible for the element-wise multiplication of two input values. In our case, it multiplies a pixel value from the input image with the corresponding kernel value.

2. **Accumulator**: The accumulator component accumulates the results of the multiplications. It maintains the running sum of the products obtained from the multiplier.

3. **Control Logic**: The control logic coordinates the operation of the multiplier and accumulator. It ensures that multiplications and accumulations occur in the correct sequence.

The MAC unit operates as follows:

1. The input image pixel and the corresponding kernel value are provided to the multiplier simultaneously.

2. The multiplier performs the multiplication of the input values and produces a product.

3. The product is then added to the compute unit, which maintains the running sum of products.

4. The final accumulated value represents the output pixel, and it is passed to the subsequent stages of the image filtering process.

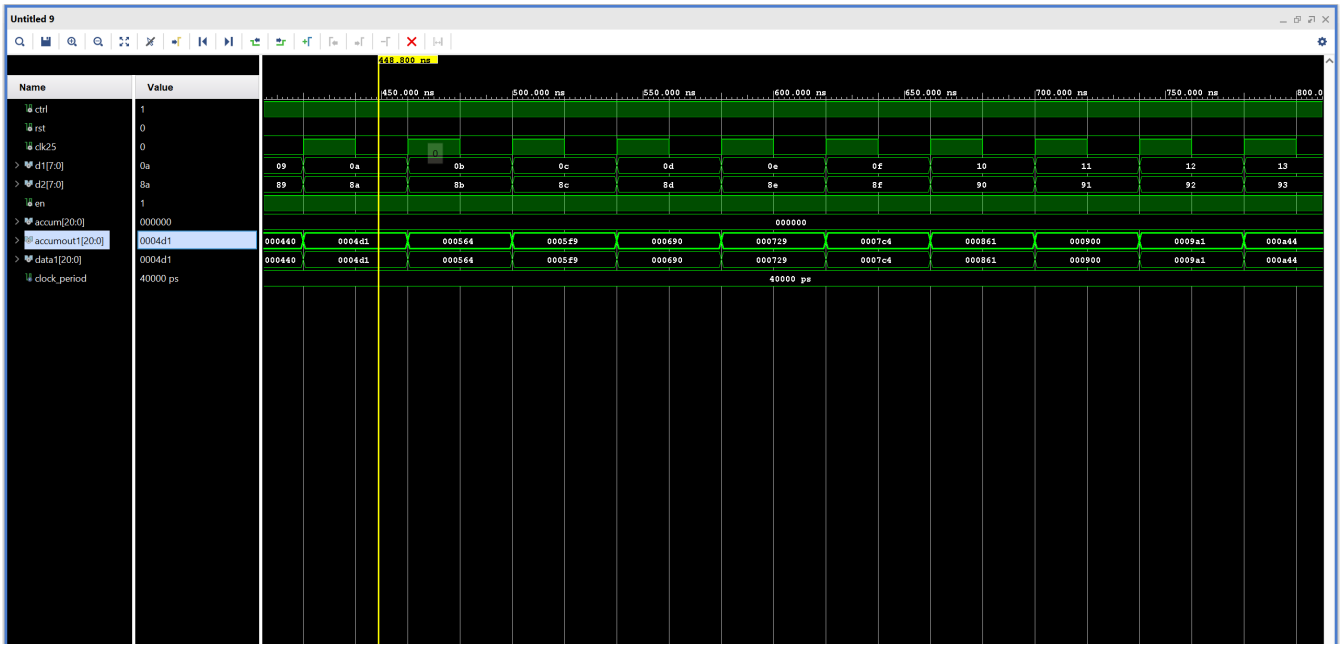
4 Testbenches and WaveForms of all Components

4.1 Final Module

TestBench is the original Image provided of coins -coins_64_bin.coe.

4.2 MAC Module

Simulation to check product of 2 random 8 bit numbers.



5 Block Diagram and Simulation