CMPE 530 Term Project

Multiply and Accumulate (MAC)

Datapath Unit Design

Mike Schroeder

Caitlin Barron

Submitted: December 9th, 2019

Lab Section: 01L2

Instructor: Sayed Ashraf Mamun

TA: Sabrina Ly

Andrew Fountain

Lecture Section: 01

Professor: Amlan Ganguly

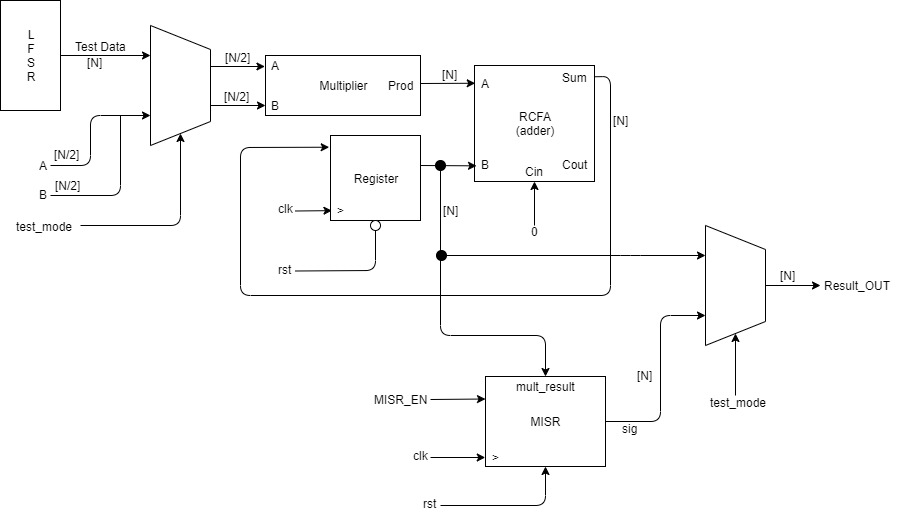
1. **Abstract**
2. **Design Methodology**
3. **Results and Analysis**
4. **Conclusion**
5. **Appendix**

**Abstract**

The objective of this exercise was to examine the full design of a Multiply and Accumulate (MAC) unit. MAC units are very useful for digital signal processing, linear algebra and more. The MAC was first built using VHDL code, which was then used to generate a schematic and layout. One of the key parts of this exercise is to implement a self-testing design. This was done by using a Linear Feedback Shift Register (LFSR) to generate test inputs and a Multiple-input Signature Register (MISR) to check the output.

**Design Methodology**

This exercise began by writing VHDL code for the MAC and each of its’ components. A block diagram of the complete system can be seen below.



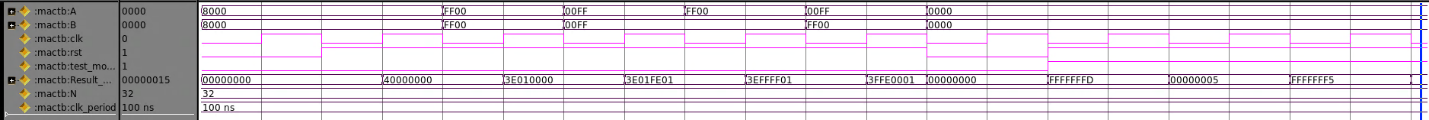
**Figure 1: A Block Diagram of the Full MAC**

Figure 1 above shows how each part of he system connects and the size of each signal. From a high level, the MAC has 5 inputs. A and B are the two values to be multiplied, they are both vectors of N/2 bits. A clock (clk) signal, reset (rst) signal, and test mode flag (test\_mode) are also passed into the MAC, these are all 1-bit signals. The MAC only has one output, the final result (Result\_OUT), which is a N-bit vector. To implement the Built In Self-Test (BIST) part of the assignment, an LFSR and MISR were included, and their signals were routed through multiplexors. When test mode is enabled, the values for A and B are taken from LFSR instead of the original values of A and B. Similarly, the final output from the MAC is the signature from the MISR as opposed to the normal output from the register.

For this design, the multiplier consists of a series of full adders that combine to create the N-bit product vector. The RCFA serves as the adder for the MAC, and similarly to the multiplier, is a series of full adders that generate the N-bit sum of the register and multiplier that gets fed back into the register. It is worth noting that in this design, the MISR has been set to only update its’ value on the falling edge of the clock to avoid a conflict with the update of the accumulator register. Fixed point math? Shift or parallel?

**Results and Analysis**

The VHDL code was simulated to verify that it’s working correctly. The figure below shows the results of the functional simulation.



**Figure 2: Functional Simulation of MAC**

As seen above, inputs A and B, both 8000, are correctly added to 40000000. Other values are then calculated. Later, the system is set to reset and the values are set back to zero as would be expected. Finally, test mode is enabled, and the LFSR and MISR generate test values.

Add layout stuff here.

**Conclusion**

Overall, this exercise proved to be successful in evaluating the design of a MAC unit. The multiplier, register, and adder work together to form the main function of the MAC. The LFSR and MISR successfully implement a self-testing functionality for the system, Add something about results

**Appendix**

Add code