Design and Simulation of a LMS-Based Adaptive FIR Filter

for Real-Time Signal Optimization using FPGA

Under the guidance of
Prof. Dr. Abhishek Kumar
Department of Electrical Engineering
Indian Institute of Technology Hyderabad

Riddhima Sen
Second Year Undergraduate (Graduating May 2027)
Department of Electrical Engineering
National Institute Of Technology Delhi

*

Acknowledgement

I would like to express my sincere gratitude to **Prof. Abhishek Kumar** at **IIT Hyderabad** for his invaluable guidance, constant encouragement, and insights throughout this project.

I am also grateful to the faculty and staff of the **Department of Electrical Engineering**, IIT Hyderabad, for providing a stimulating learning environment and necessary resources.

I would like to thank my **parents** for their unwavering support, and motivation throughout this project. Their constant belief in me has been my greatest strength.

This project, titled *"Design and Simulation of an LMS-Based Adaptive FIR Filter for Real-Time Signal Optimization using FPGA"*, has significantly improved my grasp of digital signal processing, Verilog coding, and FPGA deployment.

Riddhima Sen
Second Year Undergraduate
Department of Electrical Engineering
National Institute of Technology Delhi

*

Abstract

Adaptive filtering is a powerful technique in digital signal processing where the system parameters adjust automatically based on input signals and errors. One of the most widely used algorithms in this domain is the **LMS algorithm**, which leverages **gradient descent** to minimize the error between the desired and actual impulse response.

In this project, a **10-tap FIR filter** is used as the adaptive filter structure. FIR filters are widely used for their stability and linear phase properties, especially in real-time applications like noise cancellation, echo suppression, and communication systems.

The **objective of this project** is to implement LMS-based optimization of FIR filter coefficients using Verilog, simulate the design in a testbench, and deploy the function on an FPGA platform. The system learns to adapt the weights based on the input and desired signal, gradually minimizing the output error.

Key aspects include:

- Simulation of LMS updates using a PYNQ - Z2 board.
- Real-time coefficient adaptation for a 10-tap FIR filter.
- FPGA implementation for acceleration and verification.

This adaptive design bridges theoretical DSP concepts with hardware-based deployment, enabling real-time signal optimization using gradient descent on FPGA platforms.

*

LMS Optimization using Gradient Descent

The Least Mean Squares (LMS) algorithm is widely used in adaptive signal processing to minimize the error between a desired output and an estimated one. In this project, the optimization is performed using the **gradient descent** method, with a focus on hardware–software partitioning between the PS (Processing System) and PL (Programmable Logic) of an FPGA.

System Architecture

The overall architecture consists of the following components:

- **PL (Verilog)**: Evaluates the input function $f(x) = 10x^4 - 1$, calculates gradient , updates step size
- **PS - PYNQ (Python)**: gives 16 bit input to PL, and runs iteration
- **Communication**: AXI interface is used to read/write x values to the PL

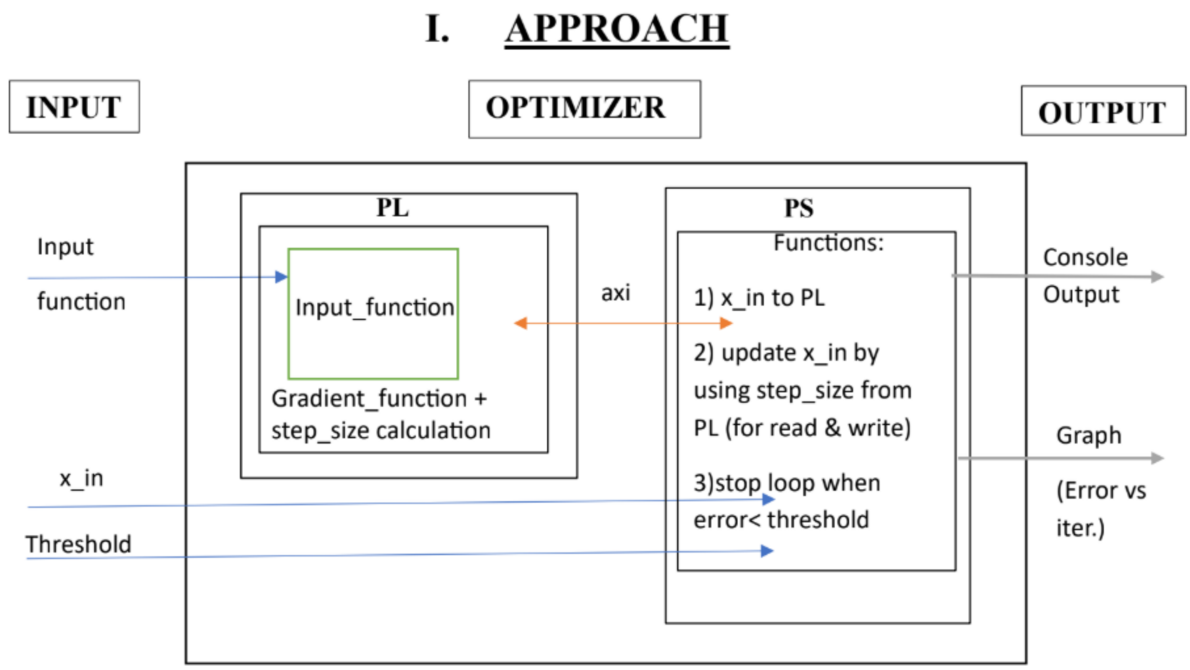

Figure 1: Block Diagram: LMS Optimization using Gradient Descent

*

Workflow

The optimization loop performs the following steps:

1. Send an initial guess of x to the PL.
2. Compute:
$$f(x+h), \quad f(x-h)$$
3. Use central difference to estimate the gradient:
$$\nabla f(x) = \frac{f(x+h) - f(x-h)}{2h}$$
4. Calculate step size:
$$\Delta x = \mu \cdot \nabla f(x)$$
5. Update x:
$$x \leftarrow x - \Delta x$$
6. Repeat until $\Delta x < \textit{threshold}$

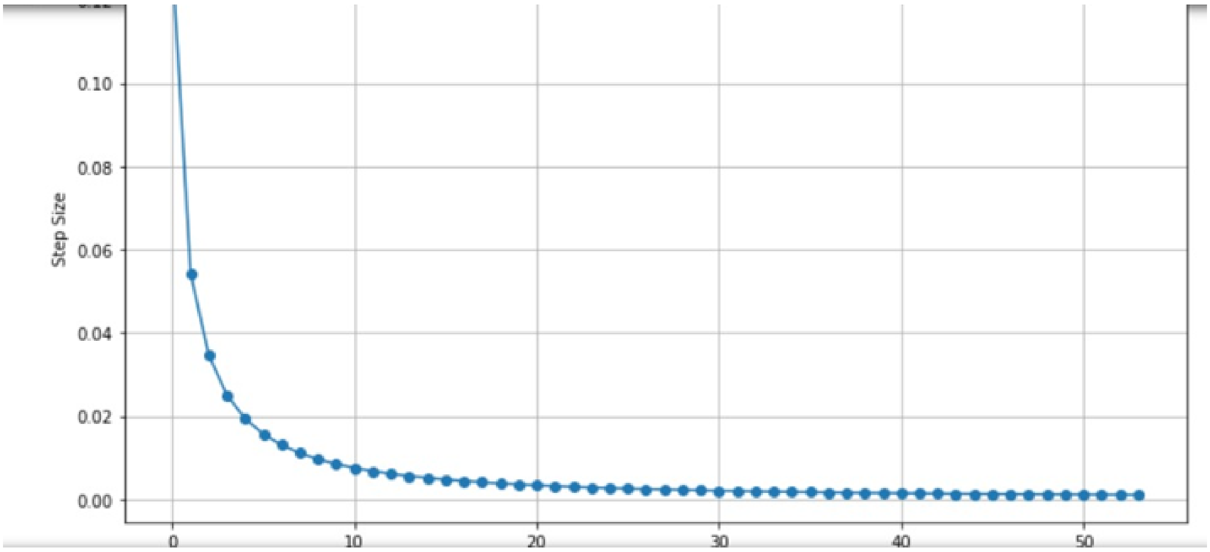

Figure 2: Error vs. Iteration Count

*

```
x = 0.500000, f(x) = -0.375000, grad = 5.199554, step = 0.129944
=== Test End ===

Iteration       x          f(x)        gradient    step size
-----------------------------------------------------------
0            0.500000   -0.375000   5.199554    0.129944
1            0.370056   -0.812500   2.174988    0.054352
2            0.315704   -0.900726   1.384644    0.034607
3            0.281097   -0.937592   1.000732    0.024994
4            0.256104   -0.957001   0.774170    0.019348
5            0.236755   -0.968628   0.625549    0.015625
6            0.221130   -0.976105   0.520874    0.013000
7            0.208130   -0.981262   0.443848    0.011078
8            0.197052   -0.984955   0.384796    0.009613
9            0.187439   -0.987671   0.338257    0.008453
10           0.178986   -0.989777   0.300873    0.007507
11           0.171478   -0.991364   0.270203    0.006744
12           0.164734   -0.992645   0.244568    0.006104
13           0.158630   -0.993683   0.222900    0.005554
14           0.153076   -0.994537   0.204590    0.005096
15           0.147980   -0.995239   0.188873    0.004700
16           0.143280   -0.995819   0.174835    0.004364
17           0.138916   -0.996307   0.162781    0.004059
18           0.134857   -0.996704   0.151947    0.003784
19           0.131073   -0.997070   0.142334    0.003540
20           0.127533   -0.997375   0.133789    0.003326
```

Figure 3: Console Output: Iteration Progression on PS

```
44           0.078522   -0.999634   0.050629    0.001251
45           0.077271   -0.999664   0.049255    0.001221
46           0.076050   -0.999695   0.047882    0.001190
47           0.074860   -0.999695   0.046509    0.001160
48           0.073700   -0.999725   0.045288    0.001129
49           0.072571   -0.999725   0.044220    0.001099
50           0.071472   -0.999756   0.043152    0.001068
51           0.070404   -0.999756   0.042084    0.001038
52           0.069366   -0.999786   0.041016    0.001007
53           0.068359   -0.999786   0.039948    0.000977

Converged (step size < threshold)!

Final x = 0.068359, f(x) = -0.999786
```

Figure 4: Console Output: Final Convergence Message

*

Adaptive FIR Filter Structure

3.1 System Overview

The Adaptive FIR Filter system is based on the LMS learning rule for optimizing filter coefficients in real time. The architecture includes the following core components:

- **Input samples** $x[n]$ stored in a register array.
- **FIR filter output** $y[n]$ computed as:

$$y[n] = \sum_{i=0}^{N-1} w_i[n] \cdot x[n-i]$$

- **Desired signal** $d[n]$ used as a reference for learning.
- **Error signal** $e[n]$ calculated as:

$$e[n] = d[n] - y[n]$$

- **LMS Update Rule:**

$$w_i[n+1] = w_i[n] + 2\mu \cdot e[n] \cdot x[n-i]$$

  where μ is the learning rate (taken as 0.025 in this simulation).

3.2 Module Breakdown (Simulation)

- **input_function.v:** gives input function (10 16bit input)
- **sum_module.v:** Part of FIR filter logic.
- **error_module.v:** Computes error.
- **coeff_update.v:** Implements LMS coefficient update logic.
- **fir_filter_function_tb.v:** Testbench module that:
  - Generates random input and desired signals.
  - Simulates adaptive filtering and coefficient updates.
  - Displays console outputs and convergence logs.

*

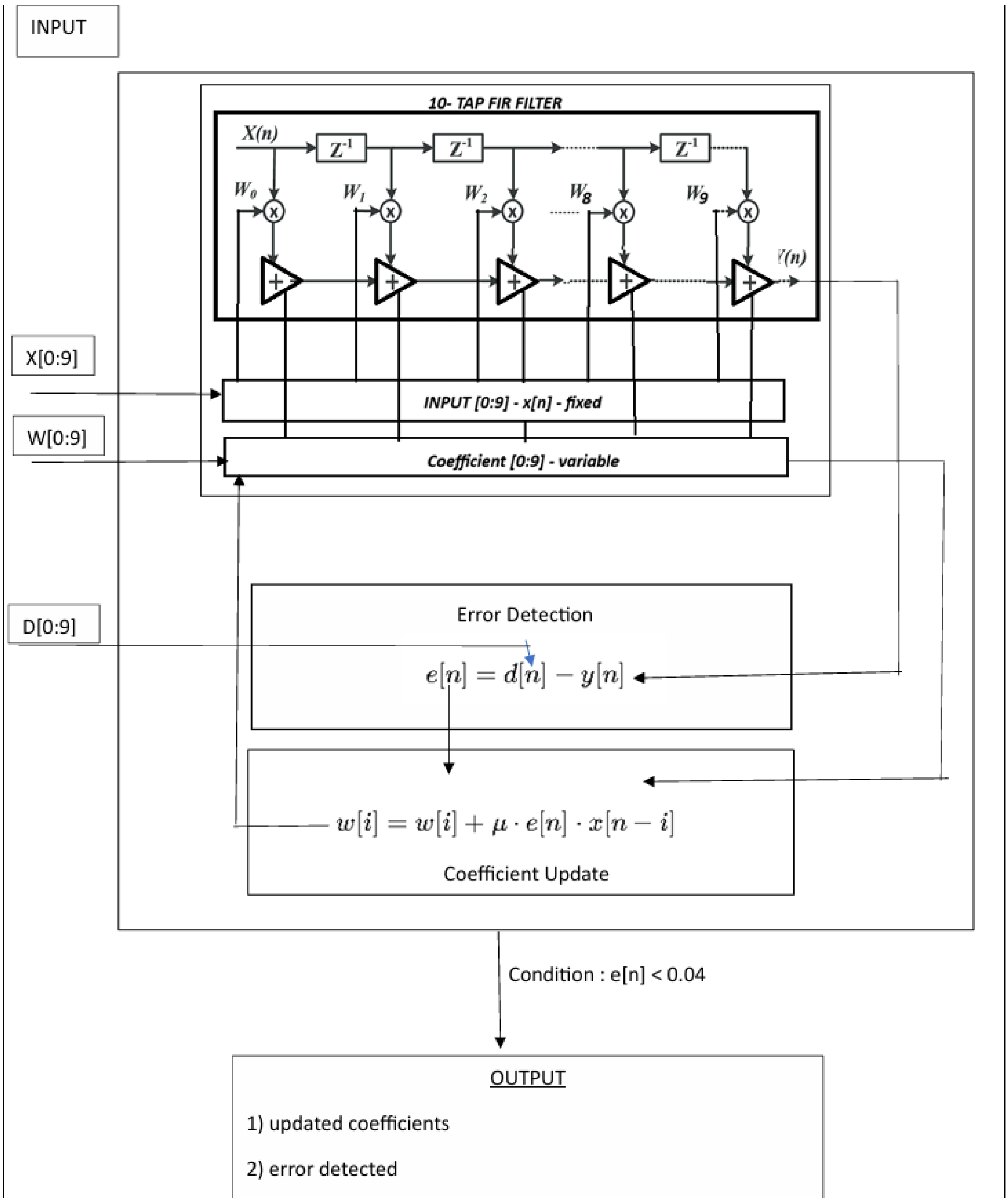

Figure 5: Block Diagram: Architecture of the Verilog LMS FIR Module

*

1) FIR FILTER – initial value

```
====== Initial Full Output Table ======
Idx:  0 | x[n]=0.500 | d[n]=0.400 | y[n]=0.250 | e[n]=0.150
Idx:  1 | x[n]=0.250 | d[n]=0.200 | y[n]=0.375 | e[n]=-0.175
Idx:  2 | x[n]=-1.000 | d[n]=0.800 | y[n]=-0.125 | e[n]=0.925
Idx:  3 | x[n]=0.500 | d[n]=0.400 | y[n]=0.125 | e[n]=0.275
Idx:  4 | x[n]=0.250 | d[n]=0.200 | y[n]=0.250 | e[n]=-0.050
Idx:  5 | x[n]=-1.000 | d[n]=0.800 | y[n]=-0.250 | e[n]=-0.950
Idx:  6 | x[n]=0.500 | d[n]=0.400 | y[n]=0.000 | e[n]=0.400
Idx:  7 | x[n]=0.250 | d[n]=0.200 | y[n]=0.125 | e[n]=0.075
Idx:  8 | x[n]=-1.000 | d[n]=0.800 | y[n]=-0.375 | e[n]=-0.825
Idx:  9 | x[n]=-1.000 | d[n]=0.800 | y[n]=-0.875 | e[n]=-0.325
Idx: 10 | x[n]=0.000 | d[n]=0.000 | y[n]=-0.875 | e[n]=0.875
```

Figure 6: Simulation Output 1

```
======= LMS Gradient Descent on Index 0 =======
 Coeff[0]=0.50000 | y=0.25000 | d=0.39999 | e=0.14999
 Coeff[0]=0.53748 | y=0.26874 | d=0.39999 | e=0.13126
 Coeff[0]=0.57028 | y=0.28513 | d=0.39999 | e=0.11487
 Coeff[0]=0.59900 | y=0.29950 | d=0.39999 | e=0.10049
 Coeff[0]=0.62411 | y=0.31204 | d=0.39999 | e=0.08795
 Coeff[0]=0.64609 | y=0.32303 | d=0.39999 | e=0.07697
 Coeff[0]=0.66531 | y=0.33264 | d=0.39999 | e=0.06735
 Coeff[0]=0.68213 | y=0.34106 | d=0.39999 | e=0.05893
 Coeff[0]=0.69684 | y=0.34842 | d=0.39999 | e=0.05157
 Coeff[0]=0.70972 | y=0.35486 | d=0.39999 | e=0.04514
 Coeff[0]=0.72098 | y=0.36047 | d=0.39999 | e=0.03952
```

Figure 7: Simulation Output 2

*

```
======= LMS Gradient Descent on Index 1 =======
 Coeff[1]=0.50000 | y=0.37500 | d=0.19998 | e=-0.17502
 Coeff[1]=0.45624 | y=0.35312 | d=0.19998 | e=-0.15314
 Coeff[1]=0.41794 | y=0.33395 | d=0.19998 | e=-0.13397
 Coeff[1]=0.38443 | y=0.31720 | d=0.19998 | e=-0.11722
 Coeff[1]=0.35510 | y=0.30255 | d=0.19998 | e=-0.10257
 Coeff[1]=0.32944 | y=0.28970 | d=0.19998 | e=-0.08972
 Coeff[1]=0.30701 | y=0.27850 | d=0.19998 | e=-0.07852
 Coeff[1]=0.28735 | y=0.26868 | d=0.19998 | e=-0.06870
 Coeff[1]=0.27017 | y=0.26007 | d=0.19998 | e=-0.06009
 Coeff[1]=0.25513 | y=0.25256 | d=0.19998 | e=-0.05258
 Coeff[1]=0.24197 | y=0.24597 | d=0.19998 | e=-0.04599
 Coeff[1]=0.23047 | y=0.24023 | d=0.19998 | e=-0.04025
 Coeff[1]=0.22040 | y=0.23520 | d=0.19998 | e=-0.03522
```

Figure 8: Simulation Output 3

```
======= LMS Gradient Descent on Index 9 =======
 Coeff[9]=0.50000 | y=-0.87500 | d=0.79999 | e=-0.32501
 Coeff[9]=0.41873 | y=-0.91565 | d=0.79999 | e=-0.28436
 Coeff[9]=0.34763 | y=-0.95120 | d=0.79999 | e=-0.24881
 Coeff[9]=0.28540 | y=-0.98230 | d=0.79999 | e=-0.21771
 Coeff[9]=0.23096 | y=0.99048 | d=0.79999 | e=-0.19049
 Coeff[9]=0.18332 | y=0.96664 | d=0.79999 | e=-0.16666
 Coeff[9]=0.14163 | y=0.94580 | d=0.79999 | e=-0.14581
 Coeff[9]=0.10516 | y=0.92758 | d=0.79999 | e=-0.12759
 Coeff[9]=0.07324 | y=0.91162 | d=0.79999 | e=-0.11163
 Coeff[9]=0.04532 | y=0.89764 | d=0.79999 | e=-0.09766
 Coeff[9]=0.02090 | y=0.88544 | d=0.79999 | e=-0.08545
 Coeff[9]=-0.00046 | y=0.87476 | d=0.79999 | e=-0.07477

Coeff[9]=-0.00046 | y=0.87476 | d=0.79999 | e=-0.07477
Coeff[9]=-0.01917 | y=0.86542 | d=0.79999 | e=-0.06543
Coeff[9]=-0.03552 | y=0.85724 | d=0.79999 | e=-0.05725
```

Figure 9: Simulation Output 4

*

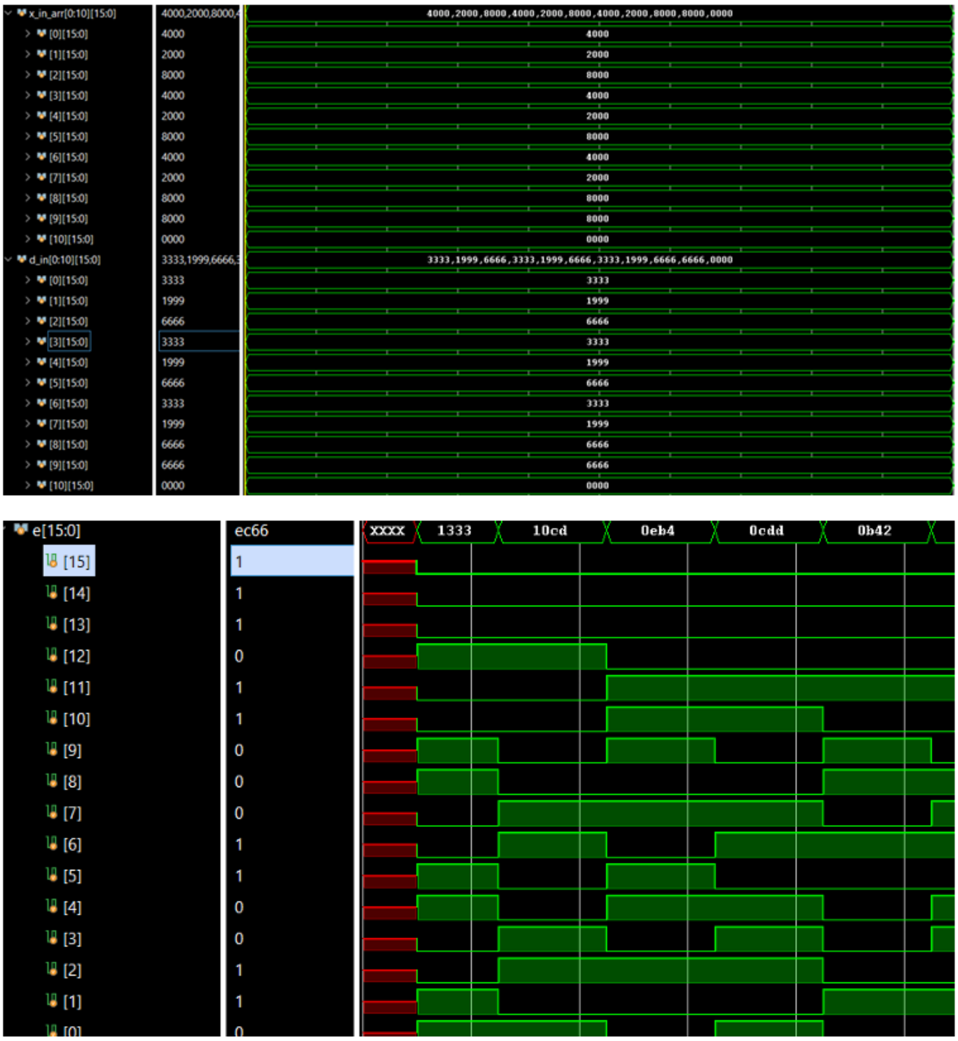

Figure 10: Simulation Output 5

Figure 11: Simulation Output 6