Lab 3: FPGA Features - Clock Management, DSP Blocks, DDR, and SRL

Written By: Joel Bailey

Date: October 9th, 2020

ECE 524/Lab

Lab 3: FPGA Features

Table of Contents

I. Introduction

In this lab, an interleaver is designed in 4 different ways to showcase multiple capabilities and features of the FPGA fabric. An interleaver takes 2 data sets and combines them into one by degrading the original signals by a common factor and adding the signals together into a single, blended stream. The equation is as follows:

Where S0 / S1 are the input signals and the alpha value is the common factor that resides between .

This lab has been broken down into 4 tasks and the remainder of the report will discuss each task separately within each section.

Task 1 is a direct implementation of an interleaver module using 2 multipliers and a single adder. Each signal and alpha factor is multiplied in the first stage and in the second, the results of multiplication are summed as seen in the following diagram:

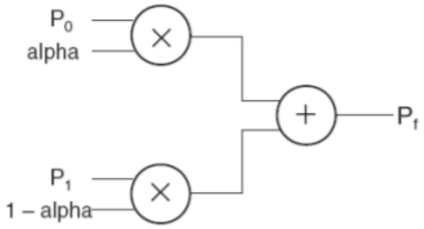

Figure 1: Task 1 implementation

Task 2 is a design implementation that removes the need for a second multiplier by clocking the arithmetic portion of the design at 2 times the clock speed. The 2x clock is implemented using the Vivado clocking wizard from the IP catalog. A high level is realized below:

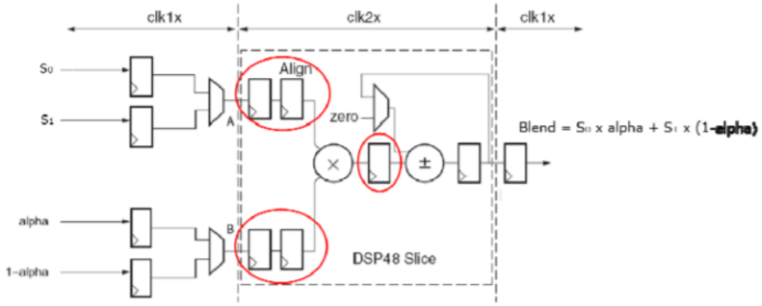

Figure 2: Task 2 with 2x clock domain

Task 3 and task 4 take the same circuit design as the previous task by adding delays to the first signal and alpha by implementing look-up tables as a shift register in 3 and BRAM in task 4.

II. Procedure

Task 1

The implementation of task one began with the calculation of the signals going into the design. As the width of each pixel is 8-bits, the normalized, signed values had to fit within that boundary 2^7 (+1 bit for sign). I chose 5Mhz sine waves where each signal was calculated from the follow equations:

Those two signals were sampled at 100MHz, giving a full period at 20 samples.

Step	ns	Signal S0	Normalized S0	Signal S1	Normalized S1
1	0	0	0	0	0
2	0.00000001	0.2781152949	35	0.1545084972	19
3	0.00000002	0.5290067271	67	0.2938926261	37
4	0.00000003	0.7281152949	93	0.4045084972	51
5	0.00000004	0.8559508647	109	0.4755282581	60
6	0.00000005	0.9	115	0.5	64
7	0.00000006	0.8559508647	109	0.4755282581	60
8	0.00000007	0.7281152949	93	0.4045084972	51
9	0.00000008	0.5290067271	67	0.2938926261	37
10	0.00000009	0.2781152949	35	0.1545084972	19
11	0.0000001	0	0	0	0
12	0.00000011	-0.2781152949	-35	-0.1545084972	-19
13	0.00000012	-0.5290067271	-67	-0.2938926261	-37
14	0.00000013	-0.7281152949	-93	-0.4045084972	-51
15	0.00000014	-0.8559508647	-109	-0.4755282581	-60
16	0.00000015	-0.9	-115	-0.5	-64
17	0.00000016	-0.8559508647	-109	-0.4755282581	-60
18	0.00000017	-0.7281152949	-93	-0.4045084972	-51
19	0.00000018	-0.5290067271	-67	-0.2938926261	-37
20	0.00000019	-0.2781152949	-35	-0.1545084972	-19

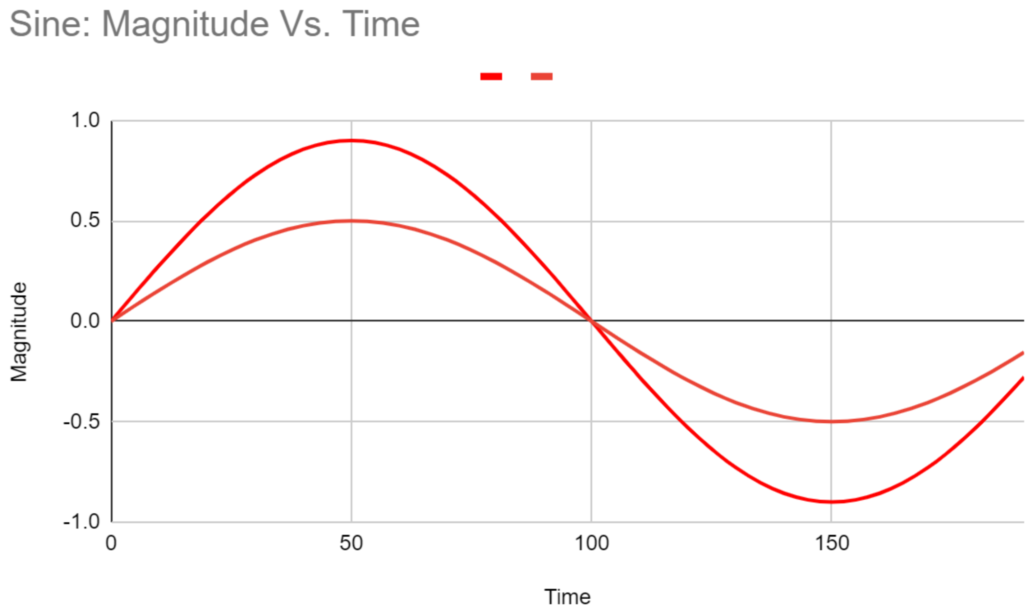

Figure 3: Input signals. 0.9Sine and 0.5Sine

The module was then coded with a clocked process that performed each operation on the rising edge.

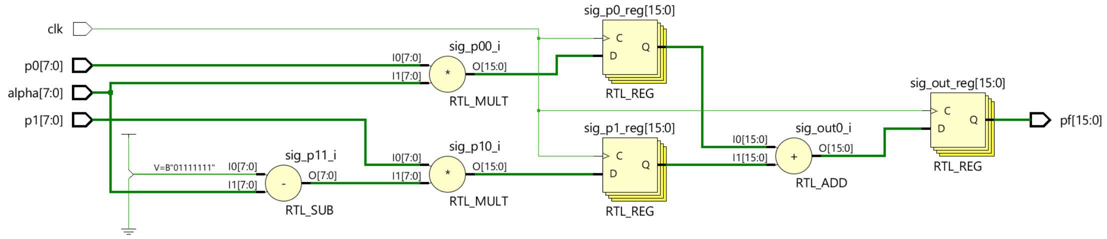

Figure 4: Task 1 elaborated design

Task 2

Task two’s implementation was two separate clock domains, clocked by the vivado clocking wizard module. The clock wizard was configured to input 50MHz and output two signals, a 1x signal at 50MHz, and a 2x signal at 100MHz. The design was then split into two modules and a top level, where each module represented a separate clock domain. Within the 1x domain, the inputs are read into registers and a multiplexer selects complementary signals using a 2x selector so both values can be read within a single, 1x clock pulse. Within the 2x domain, the complementary signals are read in, multiplied, and stored in an accumulator, to be added to the following data set. Output to pf in the 2x domain is determined by the accumulator select.

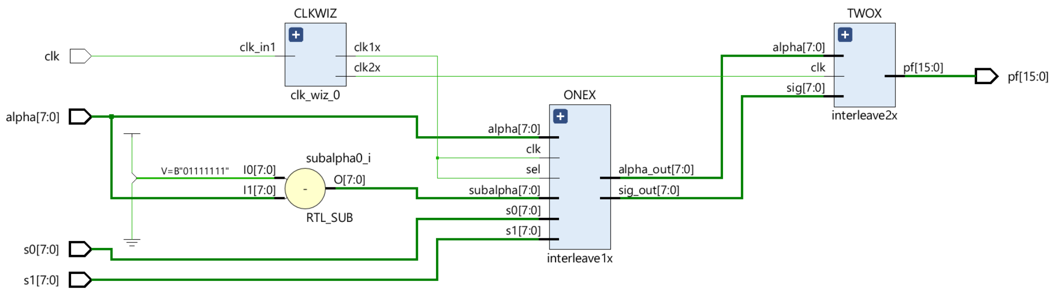

Figure 5: Task 2 elaborated design

Task 3

Signal S0 and alpha are delayed in this task using shift ram look-up tables. The only change to the lower level modules is in the 1x domain, S0 is tied directly from the delay into the multiplexer.

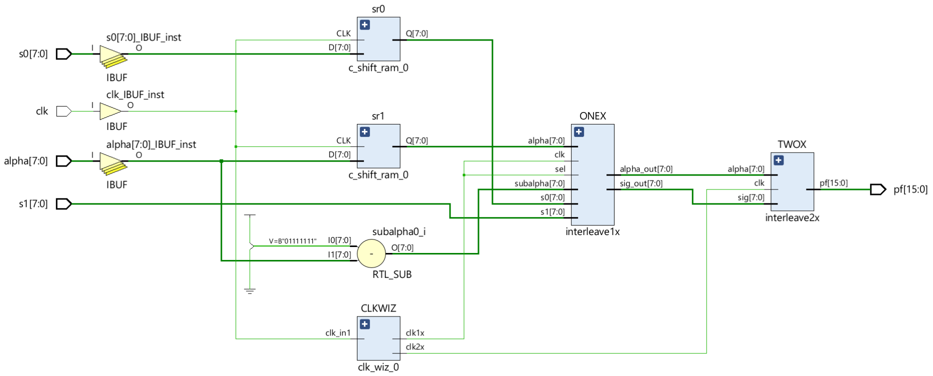

Figure 6: Task 3 elaborated design

Task 4

Task 4 mimics task 3, where a delay is added to S0 and alpha, but in this task, the delay is generated by using simple dual-port block memory. To create the delay, the inputs are written to the BRAM and subsequently read from the same address, which is looped through. In this implementation the depth of the RAM was 64-bits. Only the top level was modified to add the BRAM.

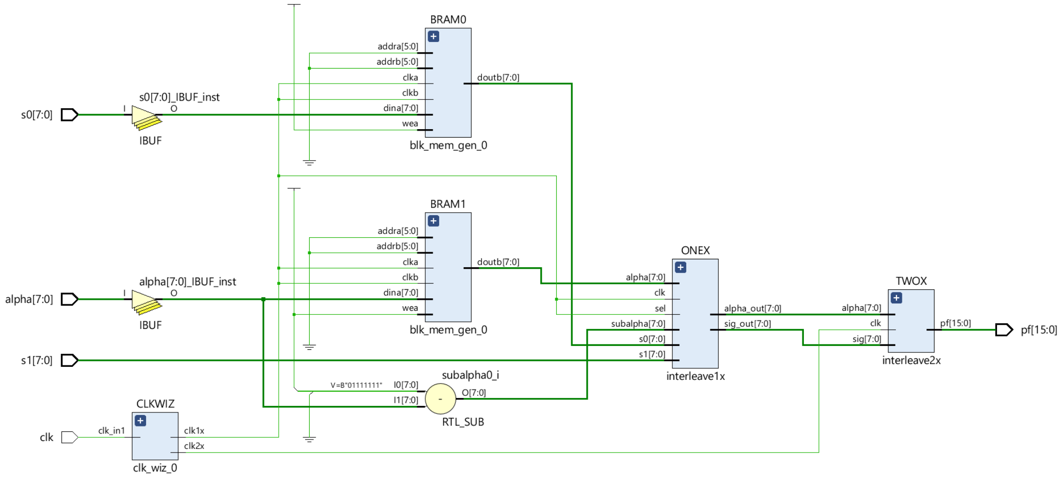

Figure 7: Task 4 elaborated design

III. Testing Strategy

Task 1

The test vectors were placed into separate files as a full period of their sampling. In the testbench, the files are opened, read till the end (a full period), then looped, to create a continuous wave. These signals were then applied to the module and the output was read into a file and graphed in Google sheets. The design was tested at 3 alpha values, 0%, 50%, and 100%.

Task 2

The same testing strategy for task 1 was used in testing task 2.

Task 3

A delay needed to be applied to the test vectors of S0 to accommodate the SRL component used in the design, so the file input was removed. In lieu of file inputs, the test vectors were coded into the testbench and run in a forever loop, with the forever loop of S0 being delayed by a determined amount. The ports were monitored during testing to ensure proper alignment of signals and realizable output. The design was tested at 4 alpha values, 25%, 50%, 0%, and 100%.

Task 4

The testing strategy applied in task 3 was also used for this task.

IV. Results (Data)

Task 1:

Time (ns)	Normalized Out	Sine Out	Normalized Out	Sine Out	Normalized Out	Sine Out
	0 alpha		50 alpha		100 alpha	
0	0	0	0	0	0	0
10	2413	0.147277832	3437	0.209777832	4445	0.2713012695
20	4699	0.2868041992	6619	0.4039916992	8509	0.5193481445
30	6477	0.395324707	9165	0.559387207	11811	0.7208862305
40	7620	0.4650878906	10756	0.6564941406	13843	0.844909668
50	8128	0.49609375	11392	0.6953125	14605	0.891418457
60	7620	0.4650878906	10756	0.6564941406	13843	0.844909668
70	6477	0.395324707	9165	0.559387207	11811	0.7208862305
80	4699	0.2868041992	6619	0.4039916992	8509	0.5193481445
90	2413	0.147277832	3437	0.209777832	4445	0.2713012695
100	0	0	0	0	0	0
110	-2413	-0.147277832	-3437	-0.209777832	-4445	-0.2713012695
120	-4699	-0.2868041992	-6619	-0.4039916992	-8509	-0.5193481445
130	-6477	-0.395324707	-9165	-0.559387207	-11811	-0.7208862305
140	-7620	-0.4650878906	-10756	-0.6564941406	-13843	-0.844909668
150	-8128	-0.49609375	-11392	-0.6953125	-14605	-0.891418457
160	-7620	-0.4650878906	-10756	-0.6564941406	-13843	-0.844909668
170	-6477	-0.395324707	-9165	-0.559387207	-11811	-0.7208862305
180	-4699	-0.2868041992	-6619	-0.4039916992	-8509	-0.5193481445
190	-2413	-0.147277832	-3437	-0.209777832	-4445	-0.2713012695

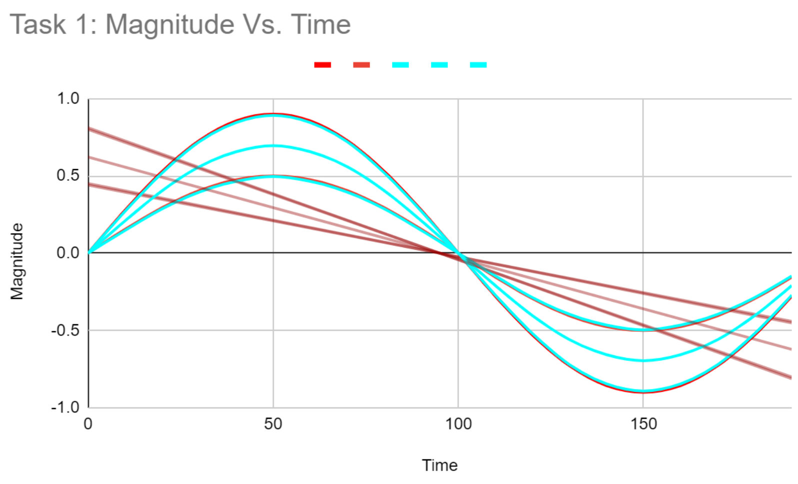

Figure 8: Results of test vectors applied to task 1's design. Each blue line represents an output at a different alpha value, 0, 50, and 100 from smallest magnitude to largest.

Task 2:

Time (ns)	Normalized Out	Sine Out	Normalized Out	Sine Out	Normalized Out	Sine Out
	0 alpha		50 alpha		100 alpha	
0	0	0	0	0	0	0
10	2413	0.147277832	3437	0.209777832	4445	0.2713012695
20	4699	0.2868041992	6619	0.4039916992	8509	0.5193481445
30	6477	0.395324707	9165	0.559387207	11811	0.7208862305
40	7620	0.4650878906	10756	0.6564941406	13843	0.844909668
50	8128	0.49609375	11392	0.6953125	14605	0.891418457
60	7620	0.4650878906	10756	0.6564941406	13843	0.844909668
70	6477	0.395324707	9165	0.559387207	11811	0.7208862305
80	4699	0.2868041992	6619	0.4039916992	8509	0.5193481445
90	2413	0.147277832	3437	0.209777832	4445	0.2713012695
100	0	0	0	0	0	0
110	-2413	-0.147277832	-3437	-0.209777832	-4445	-0.2713012695
120	-4699	-0.2868041992	-6619	-0.4039916992	-8509	-0.5193481445
130	-6477	-0.395324707	-9165	-0.559387207	-11811	-0.7208862305
140	-7620	-0.4650878906	-10756	-0.6564941406	-13843	-0.844909668
150	-8128	-0.49609375	-11392	-0.6953125	-14605	-0.891418457
160	-7620	-0.4650878906	-10756	-0.6564941406	-13843	-0.844909668
170	-6477	-0.395324707	-9165	-0.559387207	-11811	-0.7208862305
180	-4699	-0.2868041992	-6619	-0.4039916992	-8509	-0.5193481445
190	-2413	-0.147277832	-3437	-0.209777832	-4445	-0.2713012695

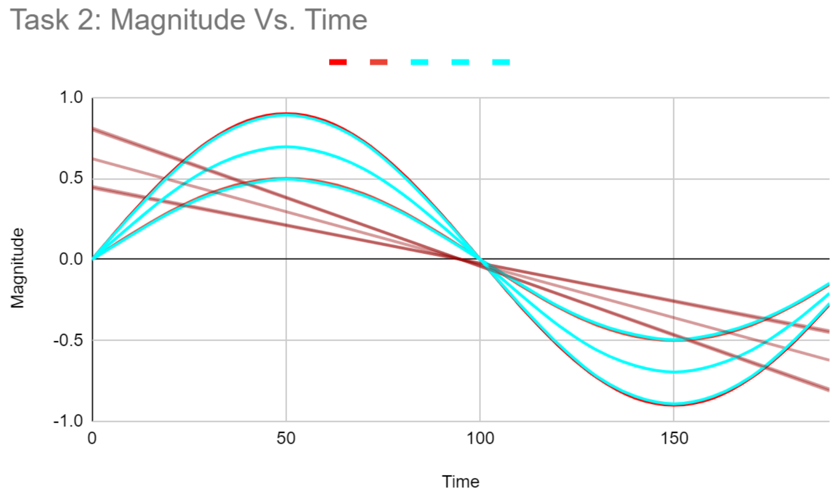

Figure 9: Results of test vectors applied to task 2's design. Each blue line represents an output at a different alpha value, 0, 50, and 100 from smallest magnitude to largest.

Task 3:

Time (ns)	Normalized Out	Sine Out	Normalized Out	Sine Out	Normalized Out	Sine Out	Normalized Out	Sine Out
	0 alpha		25 alpha		50 alpha		100 alpha	
0	0	0	0	0	0	0	0	0
10	2413	0.147277832	2925	0.178527832	3437	0.209777832	4445	0.2713012695
20	4699	0.2868041992	5659	0.3453979492	6619	0.4039916992	8509	0.5193481445
30	6477	0.395324707	7821	0.477355957	9165	0.559387207	11811	0.7208862305
40	7620	0.4650878906	9188	0.5607910156	10756	0.6564941406	13843	0.844909668
50	8128	0.49609375	9760	0.595703125	11392	0.6953125	14605	0.891418457
60	7620	0.4650878906	9188	0.5607910156	10756	0.6564941406	13843	0.844909668
70	6477	0.395324707	7821	0.477355957	9165	0.559387207	11811	0.7208862305
80	4699	0.2868041992	5659	0.3453979492	6619	0.4039916992	8509	0.5193481445
90	2413	0.147277832	2925	0.178527832	3437	0.209777832	4445	0.2713012695
100	0	0	0	0	0	0	0	0
110	-2413	-0.147277832	-2925	-0.178527832	-3437	-0.209777832	-4445	-0.2713012695
120	-4699	-0.2868041992	-5659	-0.3453979492	-6619	-0.4039916992	-8509	-0.5193481445
130	-6477	-0.395324707	-7821	-0.477355957	-9165	-0.559387207	-11811	-0.7208862305
140	-7620	-0.4650878906	-9188	-0.5607910156	-10756	-0.6564941406	-13843	-0.844909668
150	-8128	-0.49609375	-9760	-0.595703125	-11392	-0.6953125	-14605	-0.891418457
160	-7620	-0.4650878906	-9188	-0.5607910156	-10756	-0.6564941406	-13843	-0.844909668
170	-6477	-0.395324707	-7821	-0.477355957	-9165	-0.559387207	-11811	-0.7208862305
180	-4699	-0.2868041992	-5659	-0.3453979492	-6619	-0.4039916992	-8509	-0.5193481445
190	-2413	-0.147277832	-2925	-0.178527832	-3437	-0.209777832	-4445	-0.2713012695

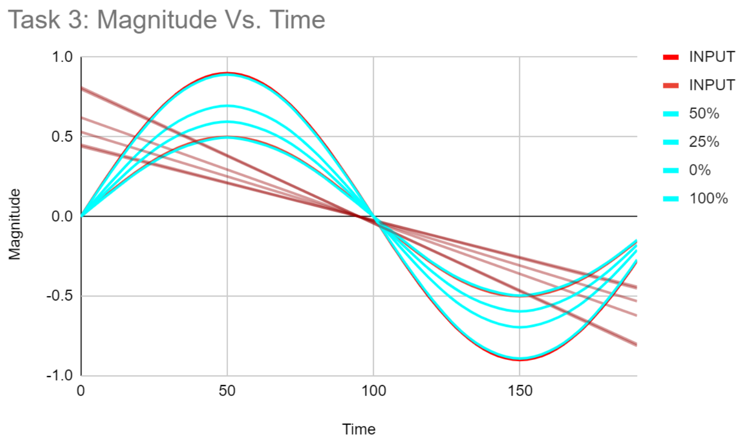

Figure 10: Results of test vectors applied to task 3's design. Each blue line represents an output at a different alpha value, 0, 25, 50, and 100.

Task 4:

Time (ns)	Normalized Out	Sine Out	Normalized Out	Sine Out	Normalized Out	Sine Out
	25 Alpha		50 Alpha		100 Alpha	
0	0	0	0	0	0	0
10	2925	0.178527832	3437	0.209777832	4445	0.2713012695
20	5659	0.3453979492	6619	0.4039916992	8509	0.5193481445
30	7821	0.477355957	9165	0.559387207	11811	0.7208862305
40	9188	0.5607910156	10756	0.6564941406	13843	0.844909668
50	9760	0.595703125	11392	0.6953125	14605	0.891418457
60	9188	0.5607910156	10756	0.6564941406	13843	0.844909668
70	7821	0.477355957	9165	0.559387207	11811	0.7208862305
80	5659	0.3453979492	6619	0.4039916992	8509	0.5193481445
90	2925	0.178527832	3437	0.209777832	4445	0.2713012695
100	0	0	0	0	0	0
110	-2925	-0.178527832	-3437	-0.209777832	-4445	-0.2713012695
120	-5659	-0.3453979492	-6619	-0.4039916992	-8509	-0.5193481445
130	-7821	-0.477355957	-9165	-0.559387207	-11811	-0.7208862305
140	-9188	-0.5607910156	-10756	-0.6564941406	-13843	-0.844909668
150	-9760	-0.595703125	-11392	-0.6953125	-14605	-0.891418457
160	-9188	-0.5607910156	-10756	-0.6564941406	-13843	-0.844909668
170	-7821	-0.477355957	-9165	-0.559387207	-11811	-0.7208862305
180	-5659	-0.3453979492	-6619	-0.4039916992	-8509	-0.5193481445
190	-2925	-0.178527832	-3437	-0.209777832	-4445	-0.2713012695

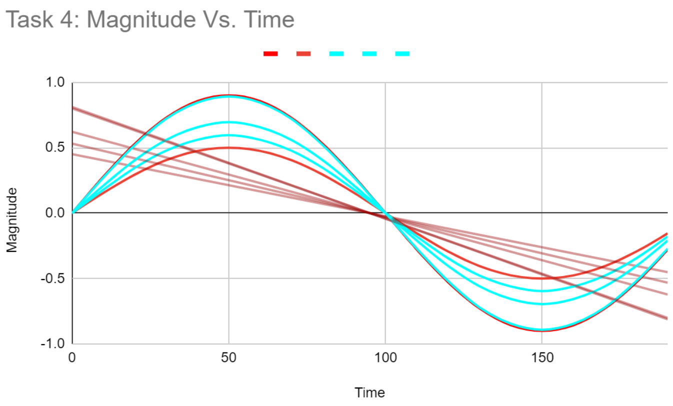

Figure 11: Results of test vectors applied to task 4’s design. Each blue line represents an output at a different alpha value, 25, 50, and 100.

V. Analysis / Conclusion

The results of task 1 was as anticipated, where the output was delayed by 2 clock pulses due to the operation being in two stages of a 1x clock. When the second time domain was added, the clocking wizard was used to generate the 2x and 1x clock. It was found that the clock generated from the clocking wizard took 740ns to engage both 1x and 2x clock signals. The clocking wizard gave the possibility to add a `locked` output to the clock that engages when the clocks are synchronized. In future designs, it would be beneficial to use the locked output to start the system. It is also worth noting that the clock implemented from the clocking wizard sends a high pulse to its outputs at the beginning of operation. This caused issues in the components that were clocked with them and not using `locked` as it gave a single clock pulse to operate on.

When a delay was added to the design in tasks 3 and 4 the components worked as expected, and because there were 20 samples per period of the sine wave the output of the 64-bit delay was offset by a 4-bit delay. This delay was added to the input by changing the input strategy from file input to hardcoded vectors. An input delay was then applied at the testbench level to the S0 signal to align the two input signals with the delay. The same was done when the BRAM was utilized which produced the proper linear relationship between the alpha value and the output signal.

Additional Questions:

1. Task 1 utilization
    a. Overview (full report found on pg 32 of the appendix)

```
2. Slice Logic Distribution
---------------------------

+-------------------------------------------+------+-------+-----------+-------+
|                 Site Type                 | Used | Fixed | Available | Util% |
+-------------------------------------------+------+-------+-----------+-------+
| Slice                                  |   44 |     0 |     13300 |  0.33 |
|   SLICEL                               |   21 |     0 |           |       |
|   SLICEM                               |   23 |     0 |           |       |
| LUT as Logic                           |  138 |     0 |     53200 |  0.26 |
|   using O5 output only                 |    0 |       |           |       |
|   using O6 output only                 |   86 |       |           |       |
|   using O5 and O6                      |   52 |       |           |       |
| LUT as Memory                          |    0 |     0 |     17400 |  0.00 |
|   LUT as Distributed RAM               |    0 |     0 |           |       |
|   LUT as Shift Register                |    0 |     0 |           |       |
| Slice Registers                        |   48 |     0 |    106400 |  0.05 |
|   Register driven from within the Slice  |   48 |       |           |       |
|   Register driven from outside the Slice |    0 |       |           |       |
| Unique Control Sets                    |    1 |       |     13300 | <0.01 |
+------------------------------------------+------+-------+-----------+-------+
* * Note: Available Control Sets calculated as Slice * 1, Review the Control Sets Report for more information
regarding control sets.


3. Memory
---------

+----------------+------+-------+-----------+-------+
|    Site Type   | Used | Fixed | Available | Util% |
+----------------+------+-------+-----------+-------+
| Block RAM Tile |    0 |     0 |       140 |  0.00 |
|   RAMB36/FIFO* |    0 |     0 |       140 |  0.00 |
|   RAMB18       |    0 |     0 |       280 |  0.00 |
+----------------+------+-------+-----------+-------+
* Note: Each Block RAM Tile only has one FIFO logic available and therefore can accommodate only one FIFO36E1
or one FIFO18E1. However, if a FIFO18E1 occupies a Block RAM Tile, that tile can still accommodate a RAMB18E1


4. DSP
------

+-----------+------+-------+-----------+-------+
| Site Type | Used | Fixed | Available | Util% |
+-----------+------+-------+-----------+-------+
| DSPs      |    0 |     0 |       220 |  0.00 |
+-----------+------+-------+-----------+-------+
```

2. No they have not been mapped to the DSP block and if I wanted to map them to a DSP block, I would want to use a directive or larger bit width. I would prove it by making a test case and running a utilization report.

3. Task 2 utilization
    a. Overview (full report found on pg 36 of the appendix)

```
2. Slice Logic Distribution
--------------------------

+------------------------------------------+------+-------+-----------+-------+
|                 Site Type                | Used | Fixed | Available | Util% |
+-------------------------------------------+------+-------+-----------+-------+
| Slice                                    |   42 |     0 |     13300 |  0.32 |
|   SLICEL                                  |   17 |     0 |           |       |
|   SLICEM                                  |   25 |     0 |           |       |
| LUT as Logic                              |  106 |     0 |     53200 |  0.20 |
|   using O5 output only                    |    0 |       |           |       |
|   using O6 output only                    |   86 |       |           |       |
|   using O5 and O6                         |   20 |       |           |       |
| LUT as Memory                             |    0 |     0 |     17400 |  0.00 |
|   LUT as Distributed RAM                  |    0 |     0 |           |       |
|   LUT as Shift Register                   |    0 |     0 |           |       |
| Slice Registers                           |   72 |     0 |    106400 |  0.07 |
|   Register driven from within the Slice   |   32 |       |           |       |
|   Register driven from outside the Slice  |   40 |       |           |       |
|     LUT in front of the register is unused |   27 |       |           |       |
|     LUT in front of the register is used   |   13 |       |           |       |
| Unique Control Sets                       |    5 |       |     13300 |  0.04 |
+-------------------------------------------+------+-------+-----------+-------+
* * Note: Available Control Sets calculated as Slice * 1, Review the Control Sets Report for more information regarding control sets.


3. Memory
---------

+----------------+------+-------+-----------+-------+
|    Site Type   | Used | Fixed | Available | Util% |
+----------------+------+-------+-----------+-------+
| Block RAM Tile |    0 |     0 |       140 |  0.00 |
|   RAMB36/FIFO* |    0 |     0 |       140 |  0.00 |
|   RAMB18       |    0 |     0 |       280 |  0.00 |
+----------------+------+-------+-----------+-------+
* Note: Each Block RAM Tile only has one FIFO logic available and therefore can accommodate only one FIFO36E1 or one
FIFO18E1. However, if a FIFO18E1 occupies a Block RAM Tile, that tile can still accommodate a RAMB18E1


4. DSP
------

+-----------+------+-------+-----------+-------+
| Site Type | Used | Fixed | Available | Util% |
+-----------+------+-------+-----------+-------+
| DSPs      |    0 |     0 |       220 |  0.00 |
+-----------+------+-------+-----------+-------+
```

4. No they have not been mapped to the DSP block, and the answer remains the same as

5. Task 3 utilization
    a. Overview (full report found on pg 40 of the appendix)

```
1. Slice Logic
--------------

+-------------------------+------+-------+-----------+-------+
|        Site Type        | Used | Fixed | Available | Util% |
+-------------------------+------+-------+-----------+-------+
| Slice LUTs*             |  107 |     0 |     53200 |  0.20 |
|   LUT as Logic          |  107 |     0 |     53200 |  0.20 |
|   LUT as Memory         |    0 |     0 |     17400 |  0.00 |
| Slice Registers         |   73 |     0 |    106400 |  0.07 |
|   Register as Flip Flop |   57 |     0 |    106400 |  0.05 |
|   Register as Latch     |   16 |     0 |    106400 |  0.02 |
| F7 Muxes               |    0 |     0 |     26600 |  0.00 |
| F8 Muxes               |    0 |     0 |     13300 |  0.00 |
+-------------------------+------+-------+-----------+-------+
* Warning! The Final LUT count, after physical optimizations and full implementation, is typically lower. Run opt_design after
synthesis, if not already completed, for a more realistic count.


1.1 Summary of Registers by Type
--------------------------------

+-------+--------------+-------------+--------------+
| Total | Clock Enable | Synchronous | Asynchronous |
+-------+--------------+-------------+--------------+
| 0     |            _ |           - |            - |
| 0     |            _ |           - |          Set |
| 0     |            _ |           - |        Reset |
| 0     |            _ |         Set |            - |
| 0     |            _ |       Reset |            - |
| 0     |          Yes |           - |            - |
| 0     |          Yes |           - |          Set |
| 16    |          Yes |           - |        Reset |
| 0     |          Yes |         Set |            - |
| 57    |          Yes |       Reset |            - |
+-------+--------------+-------------+--------------+


2. Memory
---------

+----------------+------+-------+-----------+-------+
|    Site Type   | Used | Fixed | Available | Util% |
+----------------+------+-------+-----------+-------+
| Block RAM Tile |    0 |     0 |       140 |  0.00 |
|   RAMB36/FIFO* |    0 |     0 |       140 |  0.00 |
|   RAMB18       |    0 |     0 |       280 |  0.00 |
+----------------+------+-------+-----------+-------+
* Note: Each Block RAM Tile only has one FIFO logic available and therefore can accommodate only one FIFO36E1 or one
FIFO18E1. However, if a FIFO18E1 occupies a Block RAM Tile, that tile can still accommodate a RAMB18E1


3. DSP
------

+-----------+------+-------+-----------+-------+
| Site Type | Used | Fixed | Available | Util% |
+-----------+------+-------+-----------+-------+
| DSPs      |    0 |     0 |       220 |  0.00 |
+-----------+------+-------+-----------+-------+
```

6. Yes, the same delay can be implemented using BRAM as seen in task 4. The BRAM is addressed to the max delay needed and then the values are written and read from the BRAM.

7. Task 4 utilization
    a. Overview (full report found on pg 43 of the appendix)

```
2. Slice Logic Distribution
---------------------------
```

```
+-------------------------------------------+------+-------+-----------+-------+
|                 Site Type               | Used | Fixed | Available | Util% |
+-------------------------------------------+------+-------+-----------+-------+
| Slice                                  |   47 |     0 |     13300 |  0.35 |
|   SLICEL                               |   21 |     0 |           |       |
|   SLICEM                               |   26 |     0 |           |       |
| LUT as Logic                           |  107 |     0 |     53200 |  0.20 |
|   using O5 output only                 |    0 |       |           |       |
|   using O6 output only                 |   87 |       |           |       |
|   using O5 and O6                      |   20 |       |           |       |
| LUT as Memory                          |    0 |     0 |     17400 |  0.00 |
|   LUT as Distributed RAM               |    0 |     0 |           |       |
|   LUT as Shift Register                |    0 |     0 |           |       |
| Slice Registers                        |   73 |     0 |    106400 |  0.07 |
|   Register driven from within the Slice    |   32 |       |           |       |
|   Register driven from outside the Slice   |   41 |       |           |       |
|     LUT in front of the register is unused |   30 |       |           |       |
|     LUT in front of the register is used   |   11 |       |           |       |
| Unique Control Sets                    |    5 |       |     13300 |  0.04 |
+-------------------------------------------+------+-------+-----------+-------+
* * Note: Available Control Sets calculated as Slice * 1, Review the Control Sets Report for more information regarding control sets.


3. Memory
---------

+-------------------+------+-------+-----------+-------+
|     Site Type     | Used | Fixed | Available | Util% |
+-------------------+------+-------+-----------+-------+
| Block RAM Tile    |    1 |     0 |       140 |  0.71 |
|   RAMB36/FIFO*    |    0 |     0 |       140 |  0.00 |
|   RAMB18          |    2 |     0 |       280 |  0.71 |
|     RAMB18E1 only |    2 |       |           |       |
+-------------------+------+-------+-----------+-------+
* Note: Each Block RAM Tile only has one FIFO logic available and therefore can accommodate only one FIFO36E1 or one
FIFO18E1. However, if a FIFO18E1 occupies a Block RAM Tile, that tile can still accommodate a RAMB18E1


4. DSP
------

+-----------+------+-------+-----------+-------+
| Site Type | Used | Fixed | Available | Util% |
+-----------+------+-------+-----------+-------+
| DSPs      |    0 |     0 |       220 |  0.00 |
+-----------+------+-------+-----------+-------+
```