Hardware Test Sheet

MSc in Digital Systems Engineering
Department of Electronics
University of York

Group Project

Contents

1 Test # 1

2 Tester name: Matt Reynolds

3 Test date start: 09/07/2020

4 Test date finish: 10/07/2020

5 UUT typology (VHDL entity / ~~custom IP / subsystem~~)

6 UUT name: tremolo_topLevel.vhd

7 Hardware block design screenshot

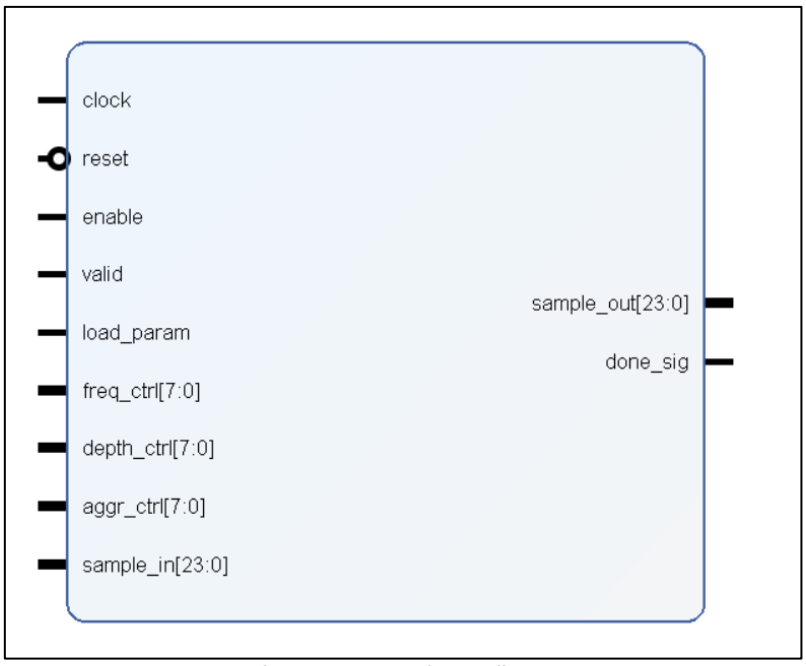

Figure 1 - Tremolo_Audio_FX

Clocked at 100 MHz. Active high reset. Enabled by user (active high). Valid, reads sample_in into input register. Load_param, updates control values. Frequency, depth, and aggression control parameters alter the modulation waveform. Sample_out contains the signed 24-bit sample for the next effect. Done signal indicates that the output is settled and ready to read.

8 Objectives of test

1. Validate timings of done_sig
2. Validate control parameters (visual inspection + measurement)
3. Validate bypass (enable) and reset state
4. Demonstrate correct response to an input tone

The following test is ran in compliance with the V-model testing strategy

9 Testbench description / test strategy overview:

The test strategy for the tremolo effect is focused on generating output data to be viewed in an x, y plot, as this is the fastest and easiest way to determine correct operation. Therefore, the test is setup such that a full modulation waveform period passes between each change of the parameters. This equates to 48000 samples.

To test the done signals and the valid signals, a sample period of ten clock cycles is used. Although in implementation, there will be over 4000 clock periods between samples, this is not practical for simulation due to the large quantities of data produced and the time it would take to compute.

The simulation uses three separate loops to test the incrementing of each parameter individually. Between these, the enable signal is toggled to determine correct operation of the bypass feature. If enable is low, the output should receive the input; bypassing modulation.

As data is plotted on a graph, running the full simulation in a single run leads to processing issues when trying to plot all the data points (the entire simulation would produce approximately 5 million samples). Therefore, I have ran it in stages, commenting out the parts that I wish to not be included. This way a single parameter can be checked at a time and reduce the overall data processing time.

Once data has been generated, I pass it into a C programme along with the reference data to generate a time base for the x-axis. This time base is relative to the 48 kHz sampling frequency that will be used by the CODEC. Following which, the data is plotted and I can observe how the input wave is modulated relative to the control parameters.

Beyond this, I use the Vivado waveform viewer to verify the control signals for the circuit. To produce an overview of the test-bench, as in Figure 13, I reduced the wave period. However, for plotted results the simulation was re-run with the appropriate time periods.

A 1 kHz test tone is used as modulation will be easy to spot at this frequency, while maintaining lower amounts of output data.

10 Test results:

Figure 2 shows the enable signal being used to bypass the modulation circuit. As it is turned off the signal (green) immediately jumps back to the input tone. When enable is set high again, the output immediately switches to the modulation output, with the address of the ROM at 0 and the start of the modulation waveform; hence the very low amplitude.

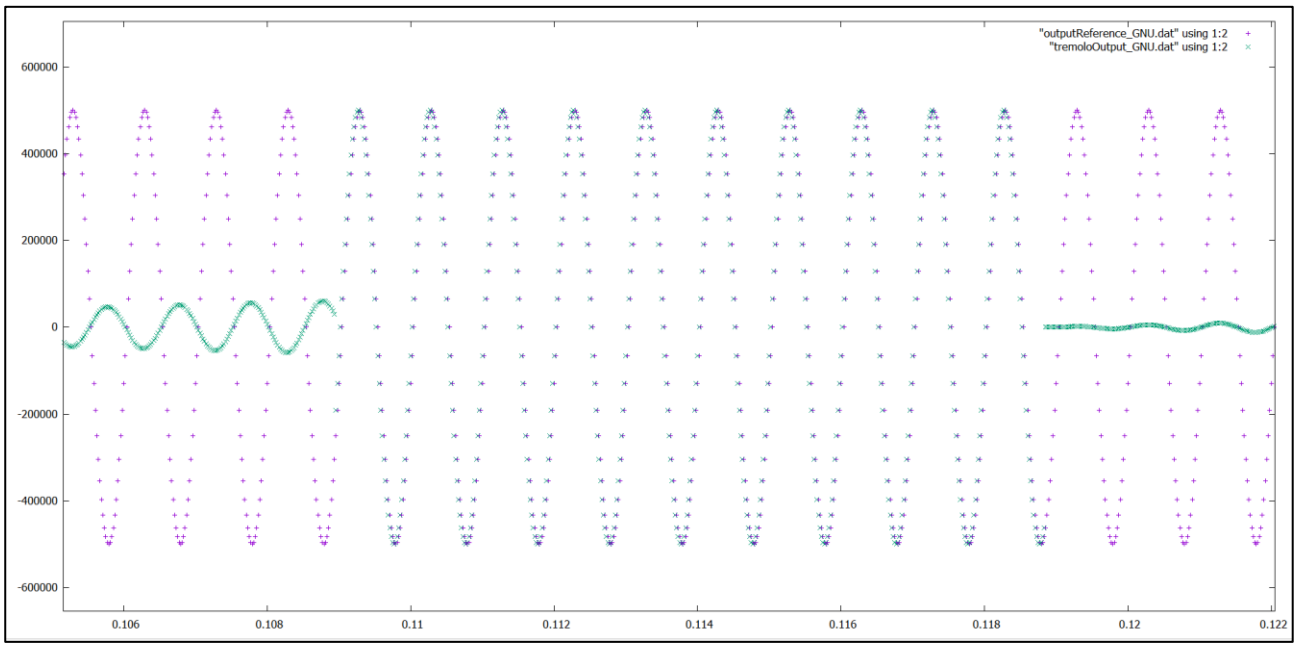

Figure 2 - Enable - GNUplot

Figure 3 clearly shows the modulation frequency increasing. At 1 Hz, a full triangle-wave is read out. At 2 Hz, we can see two triangle-wave periods occur within the wait time. This follows linearly, with 3 waves at 3 Hz, 4 at 4 Hz…and so on.

The depth control is set to 1 here, which is 0.2 of the input signal. This is why the reference (purple) is much larger than the output (green).

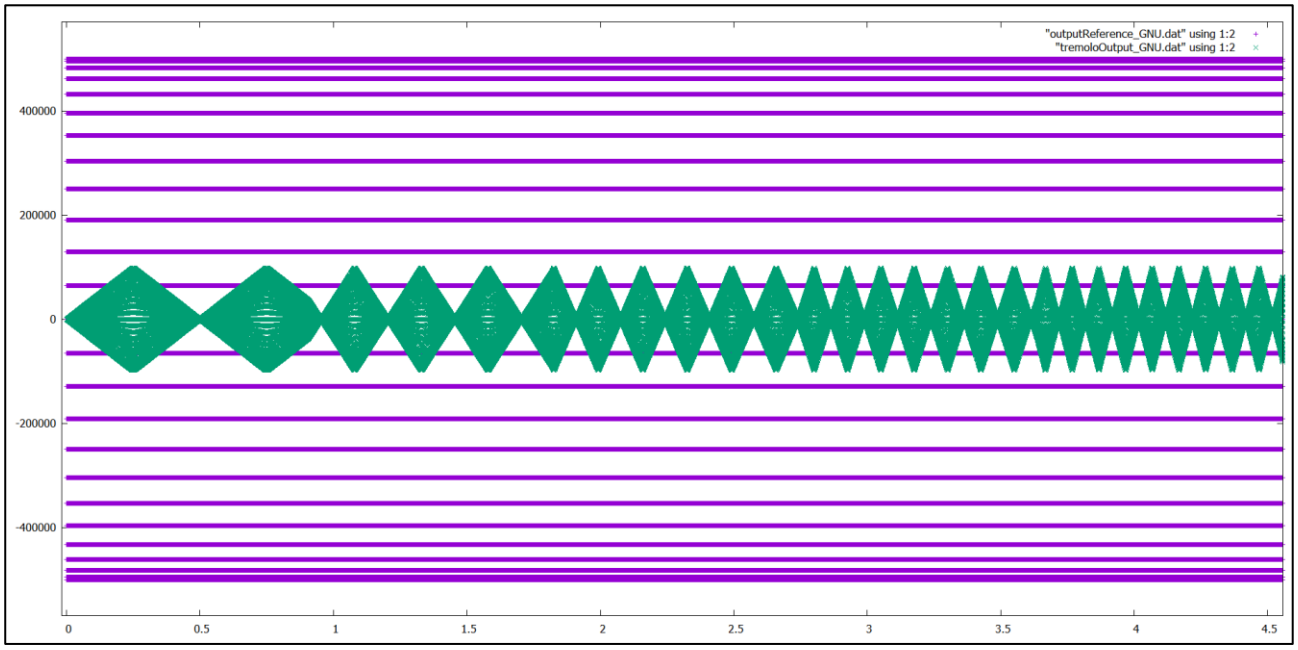

Figure 3 - Frequency control overview - GNUplot

Figure 4 gives the reader a closer look at the tremolo output signal. Here, the characteristic shape of the triangle waveform is clear, with a small amount of clipping present near the peaks of the wave; this is expected and is a result of the aggression control.

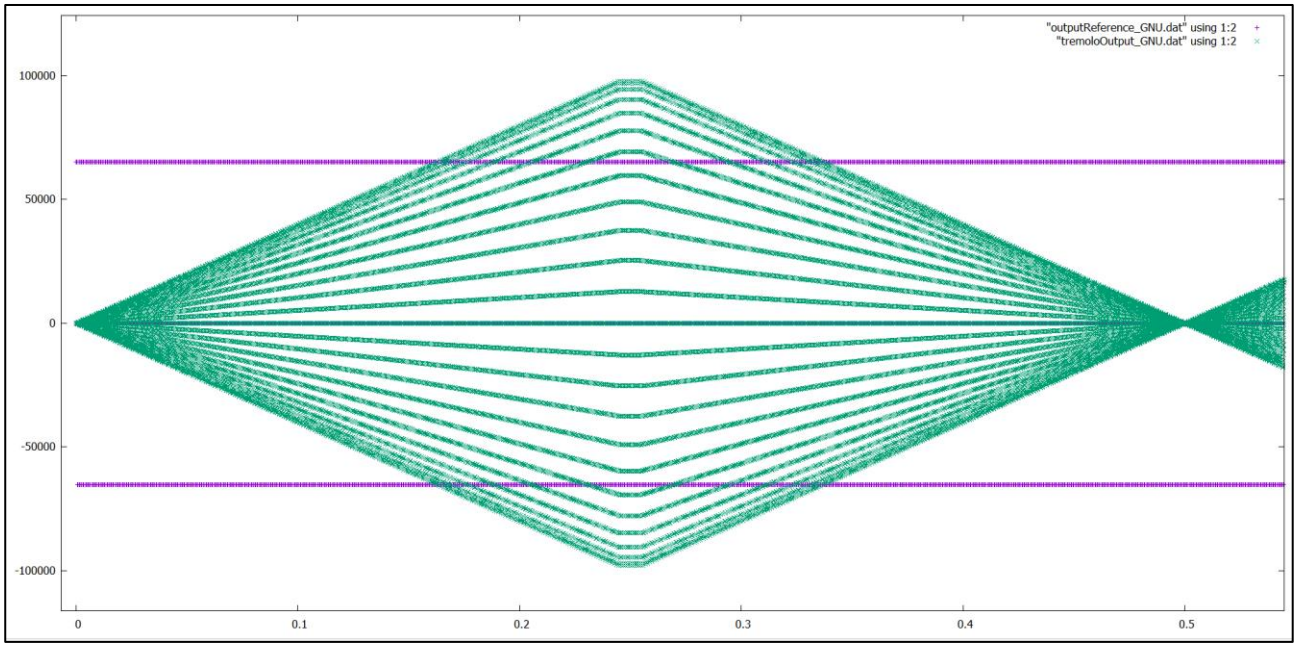

Figure 4 - One modulation period - GNUplot

Although the shape is triangular, it is difficult to identify the wave period from this level. Figure 5 offers an even closer look at the modulated signal, with two datapoints plotted at the peaks to highlight their values. We can see from these samples that the input wave's period is preserved:

0.1822 - 0.1812 = 0.001s = 1 kHz.

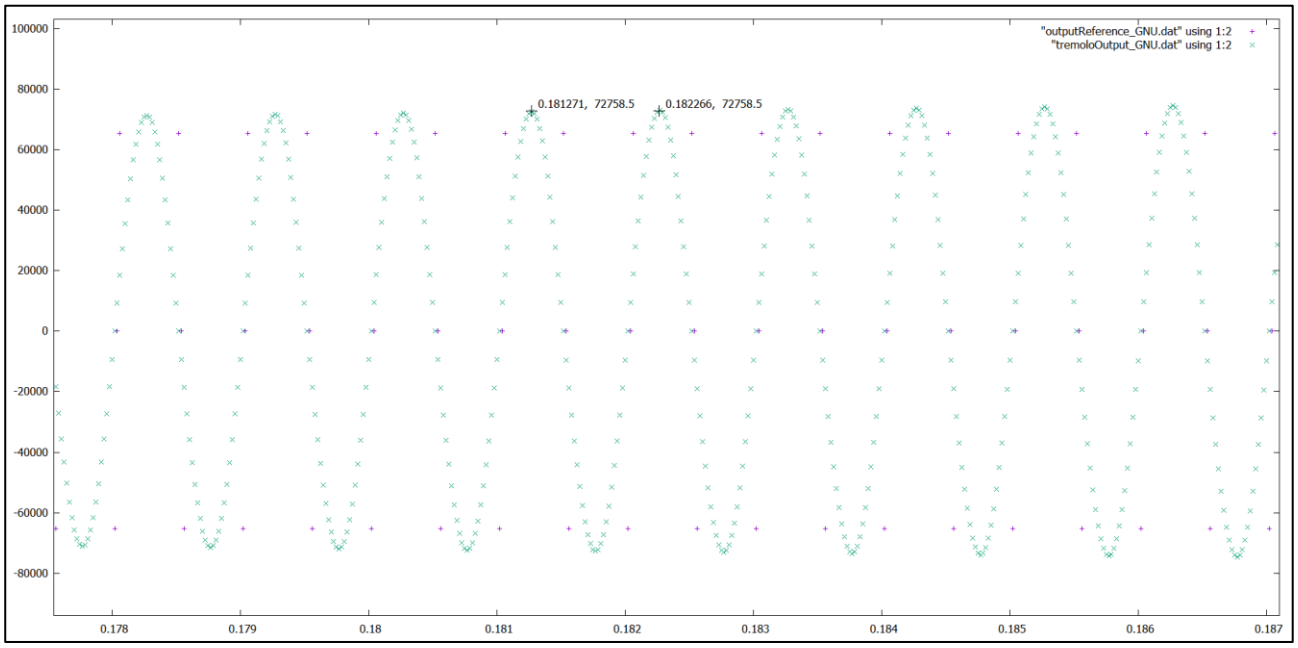

Figure 5 - Frequency preserved - GNUplot

Figure 6 allows the reader to see the comparison in peak amplitude between the input signal and the output signal, when the depth is at its minimum value:

*500'000 * 0.2 = 100'000*

Depth has a range from 1 to 10 and is scaled by 0.2.

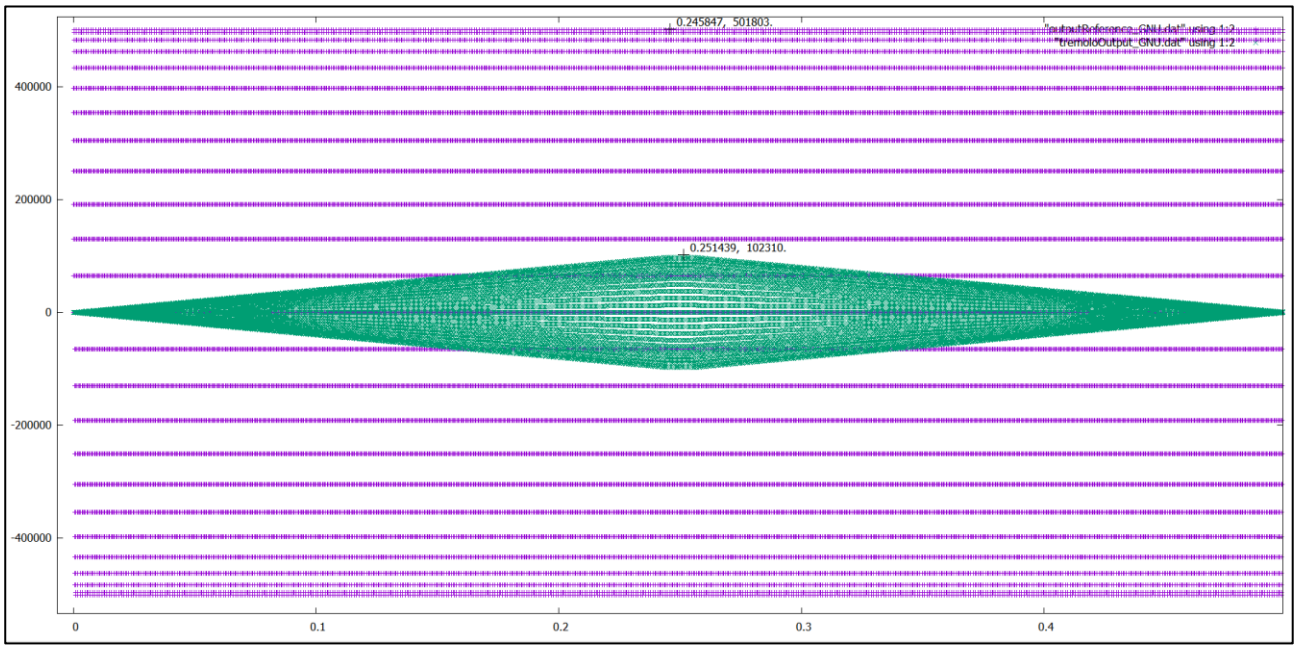

Figure 6 - Minimum depth - GNUplot

Figure 7 shows the result of increasing the depth value of the modulation waveform.

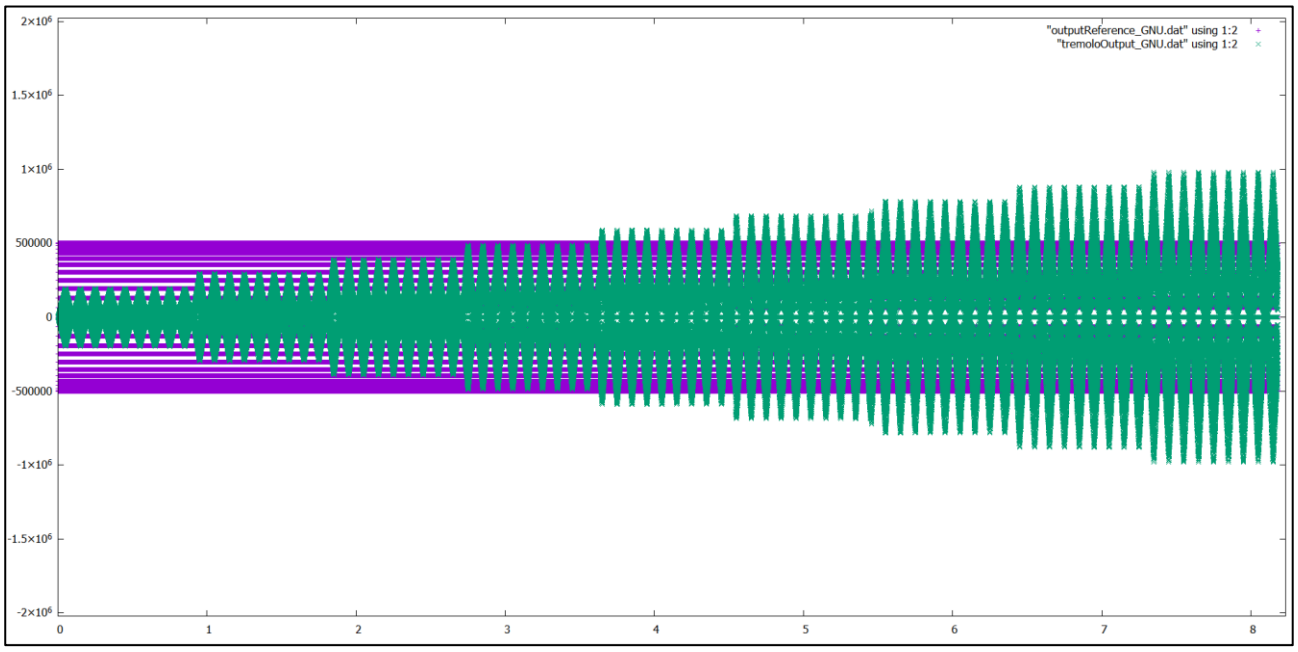

Figure 7 - Depth variation - GNUplot

Figure 8 allows us to see this more closely, with the first depth visible having the value 2 and the last depth visible, having the value 5. It can be seen from this plot that, a depth of 5 produces a unity modulation waveform, also known as 100% modulation.

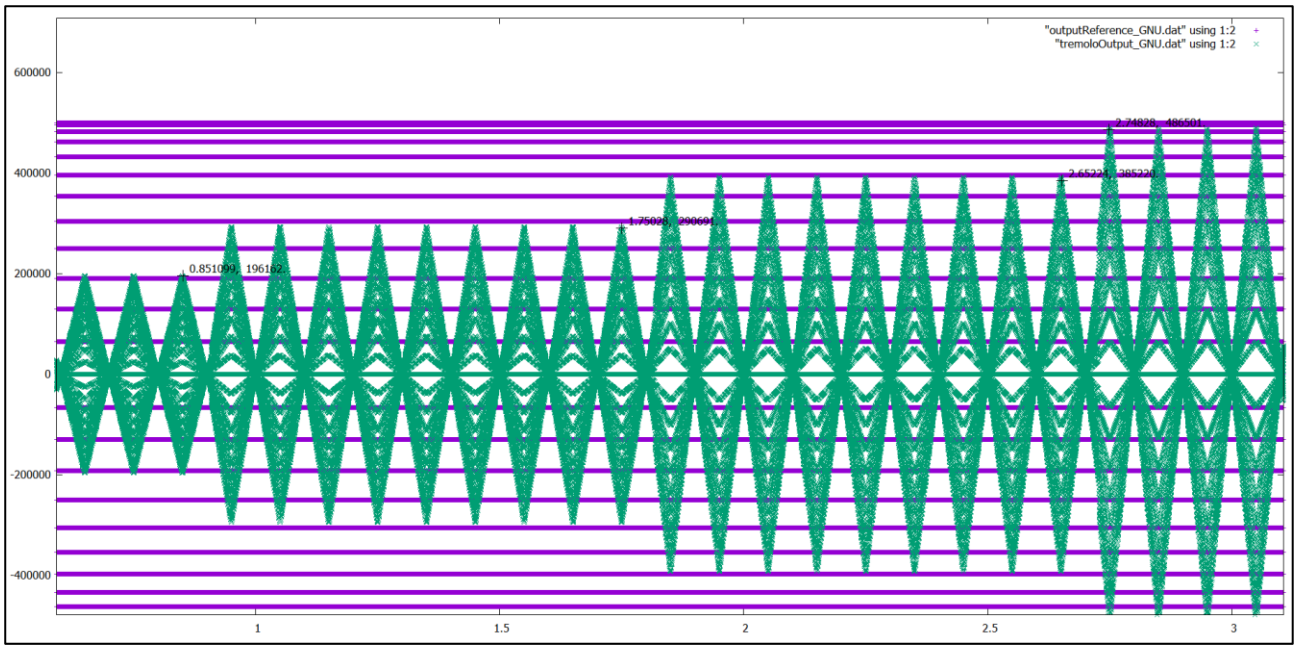

Figure 8 - Depth: 2 to 5. Aggr: 1. Freq: 5. - GNUplot

Figure 9 is an ideal graphical representation of the depth change midway through the triangle period. The moment the value is updated, the modulation output increases. However, frequency is unchanged, and the clipping remains the same.

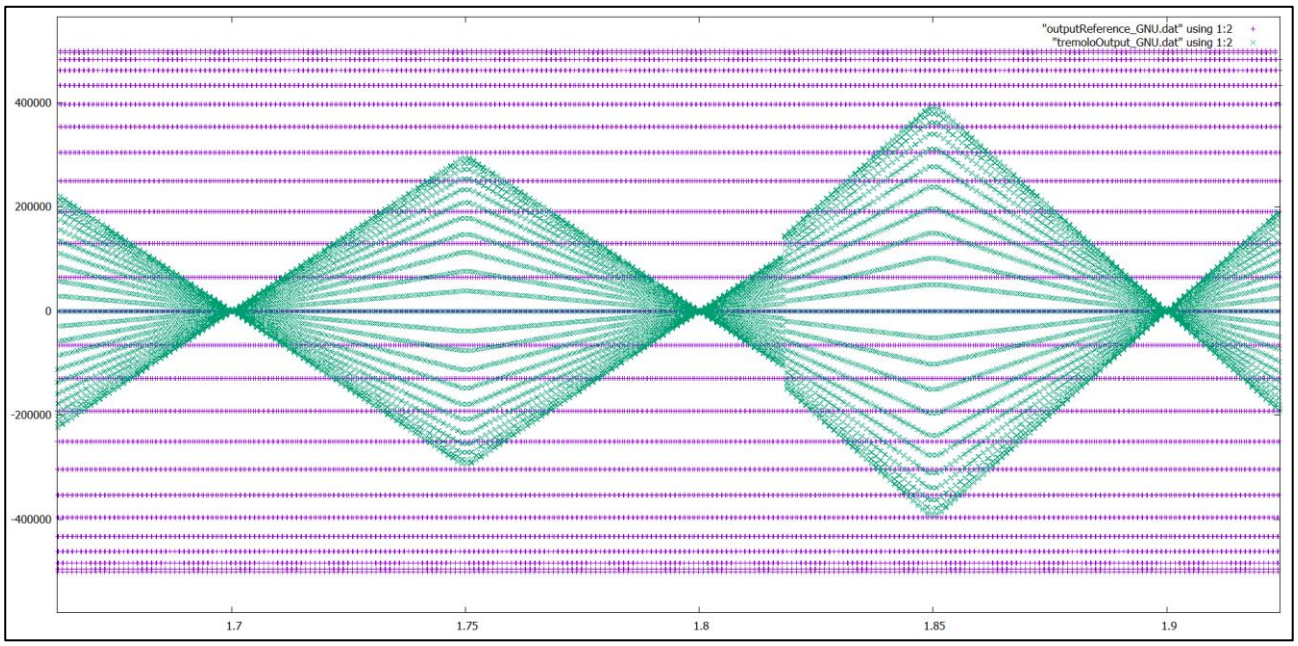

Figure 9 - Depth transition - GNUplot

Finally, aggression control is tested. Increasing the aggression, increases the amount of clipping on the modulation waveform. This produces the squared tops seen in Figure 10.

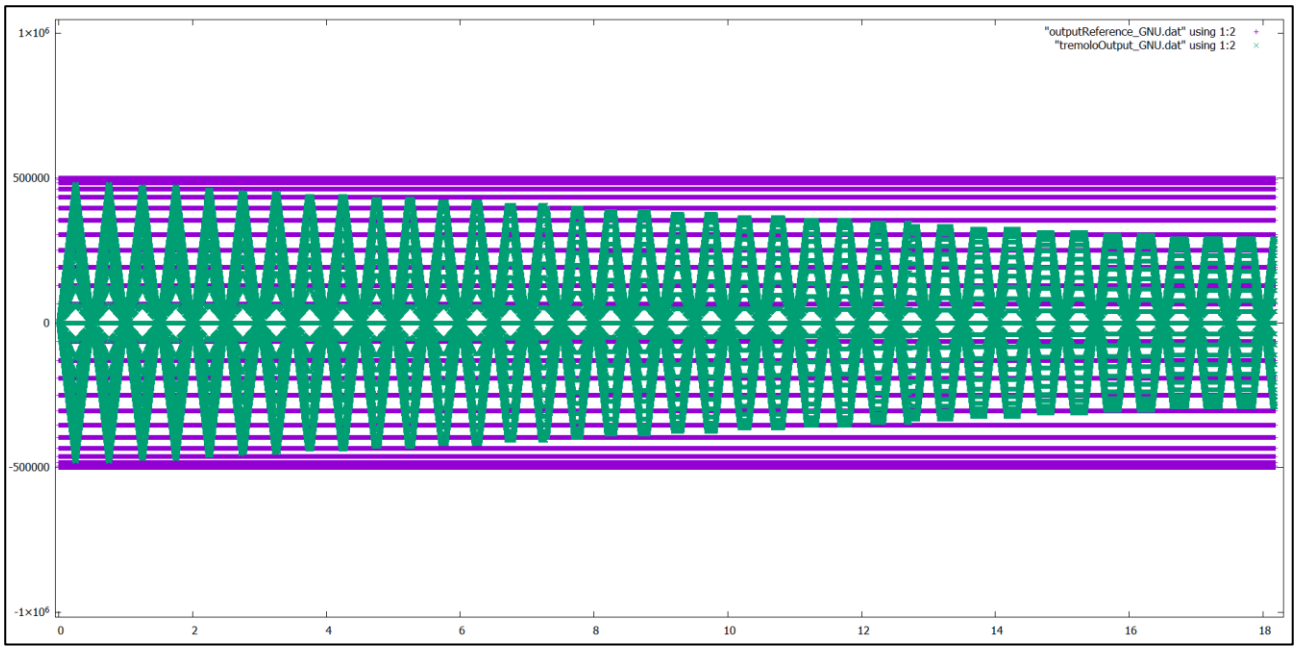

Figure 10 - Aggression control - GNUplot

Figure 11 shows a close-up of a more heavily-clipped modulation. By clipping the modulation waveform in this way it produces a more dramatic rate of change in the amplitude, which provides a more *aggressive* effect.

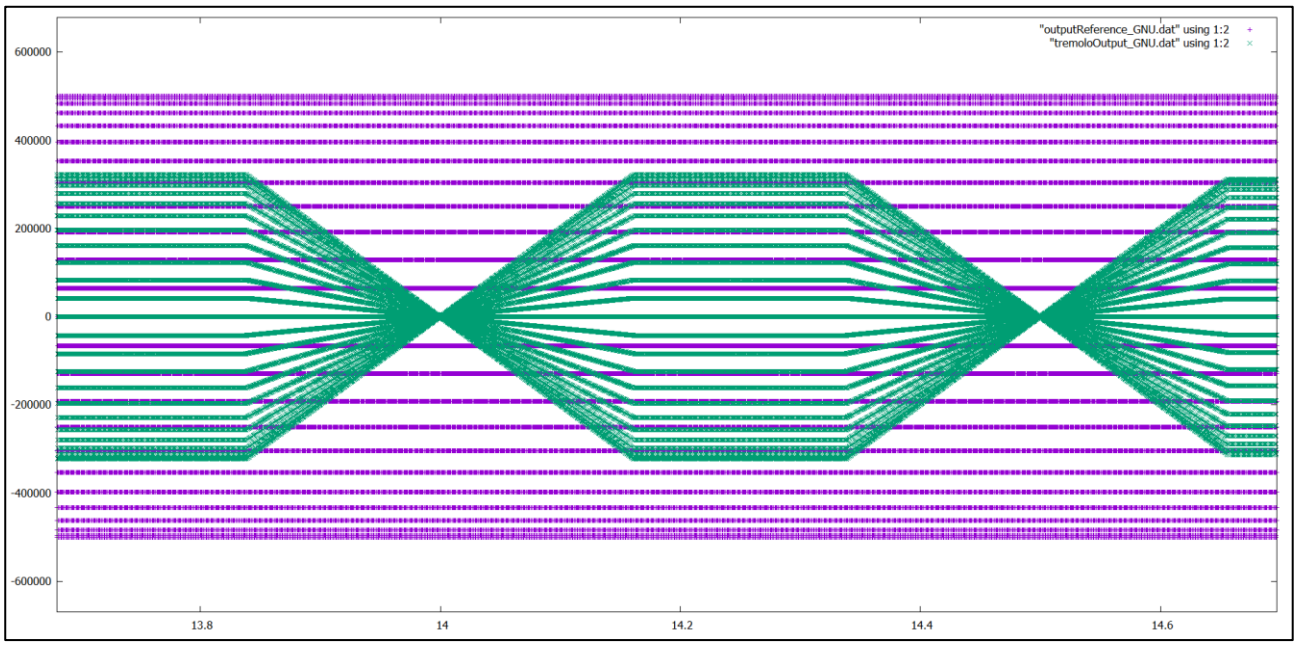

Figure 11 - Clipped modulation - GNUplot

Although the modulation waveform is clipped, no distortion is inflicted on the output signal as a result of this. This can be seen in Figure 12, where I have focussed in on the clipped modulation at T = 14.2 ms, to show the reader that the output waveform is still sinusoidal. Notice that 14.2 ms in Figure 11 is in the middle of the flat-top.

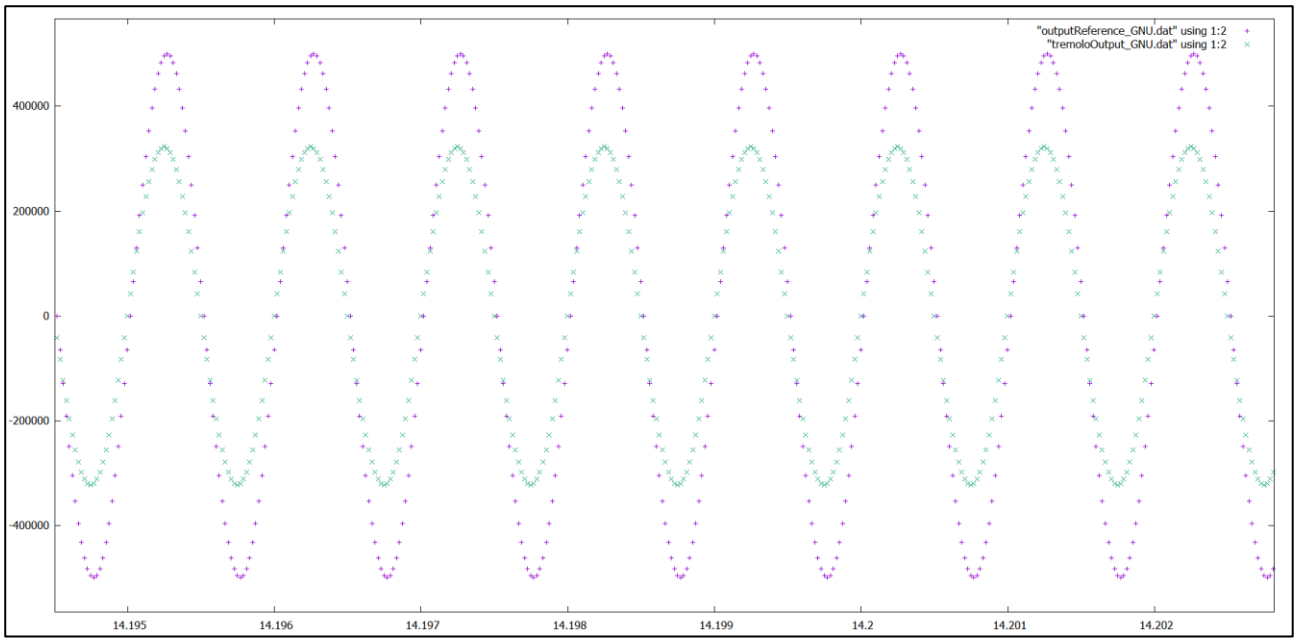

Figure 12 - T: 14.2, no distortion - GNUplot

The results show that all of the control parameters are working correctly. The control signal observations in **Section 11** show that the circuit timings are correct, according to the design.

11 Observations:

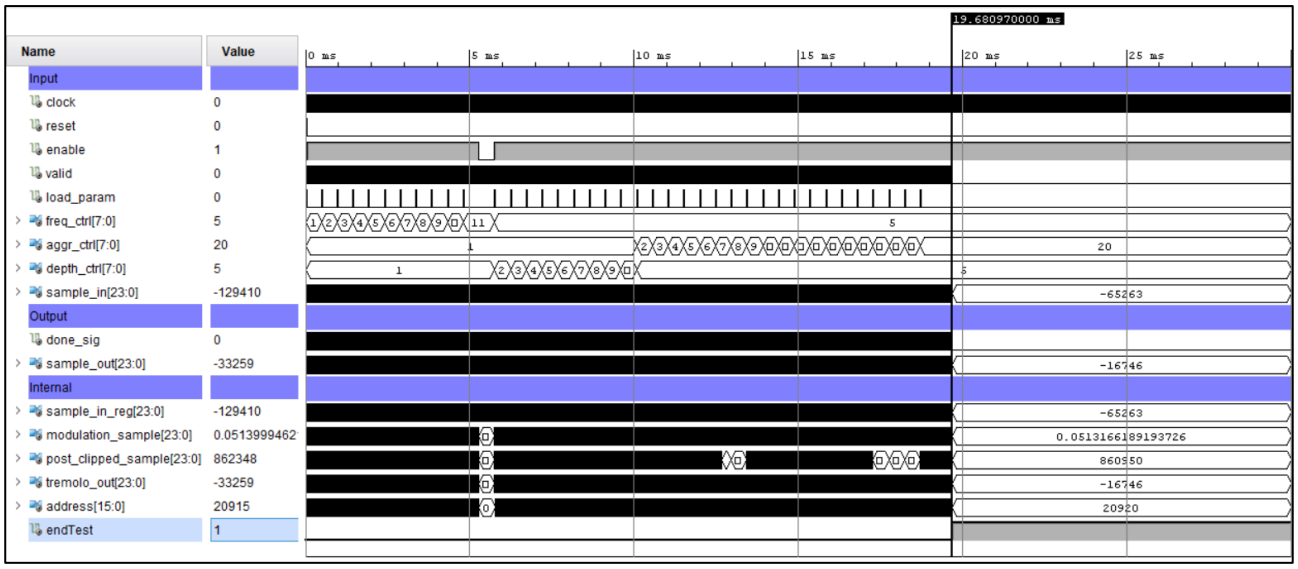

Figure 13 - Waveform overview

From Figure 13 it can be seen that the DSP is paused when the enable is low, while the output continues to change (at the same rate as the input). This is the benefit of true bypass, as power is saved by stopping the DSP circuits while the effect is disabled, preventing losses due to switching.

This also gives an overview of the test, allowing the reader to see how the control parameters are varied. When the test ends and the input samples stop being passed in, the output remains stable.

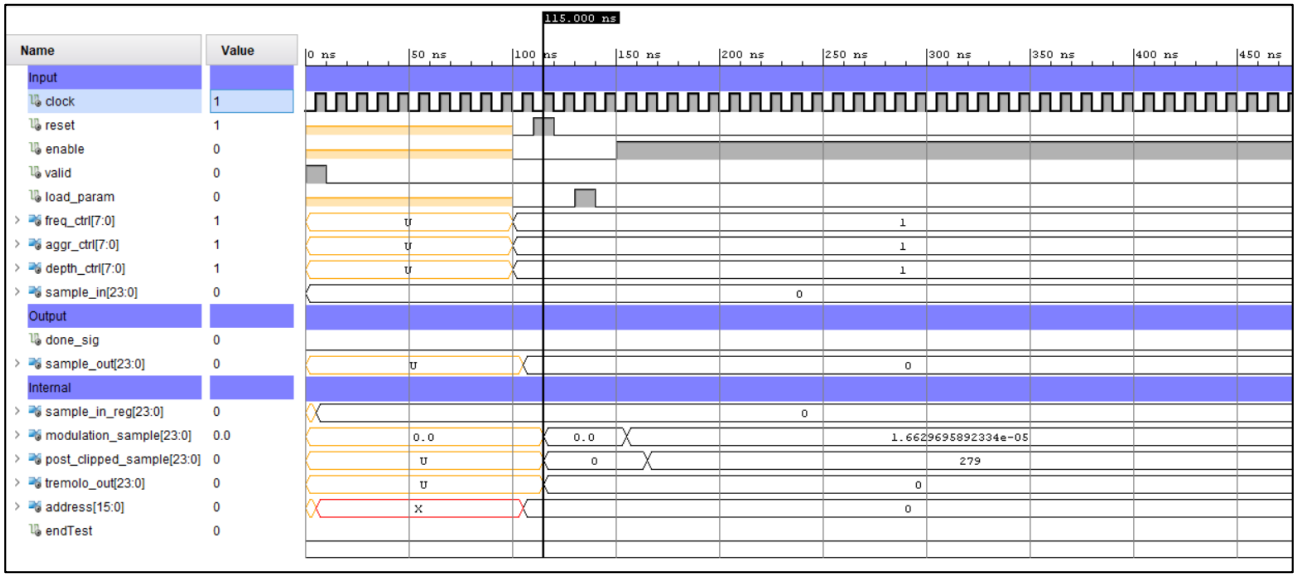

Figure 14 - Reset signal

Figure 14 shows that when the reset signal is read in, the circuit enters its reset state and sets all register values to 0.

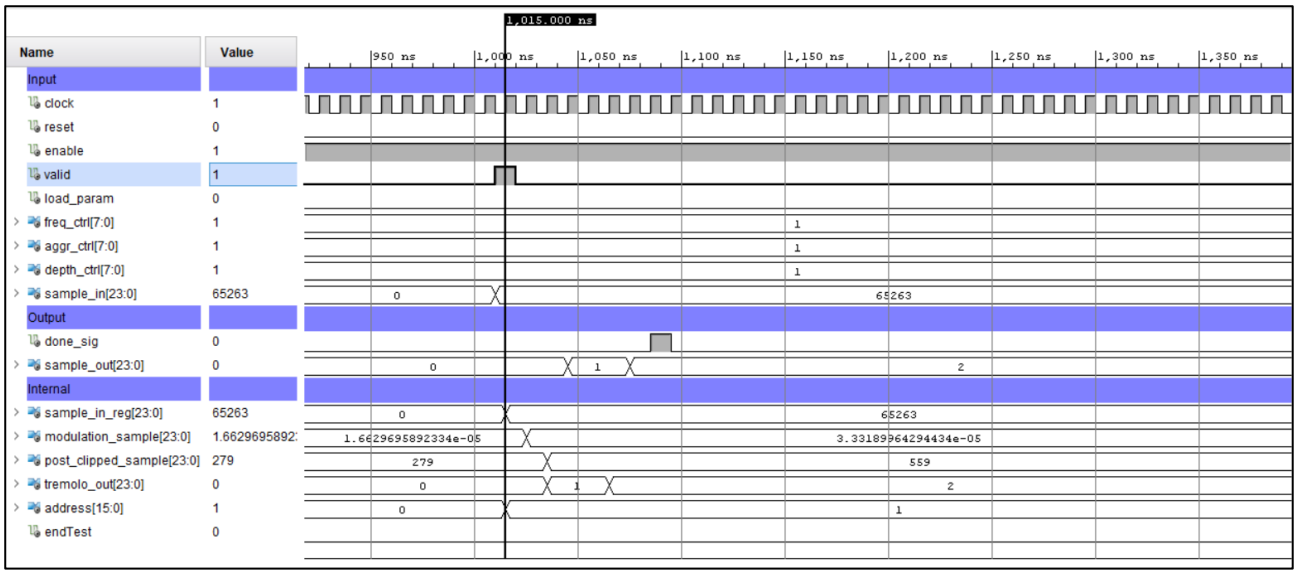

Figure 15 - Valid signal

Figure 15 shows that the samples are read into the input register when the valid signal is high and not before. The input register than holds that value, as seen by *sample_in_reg* remaining constant following the *valid* signal.

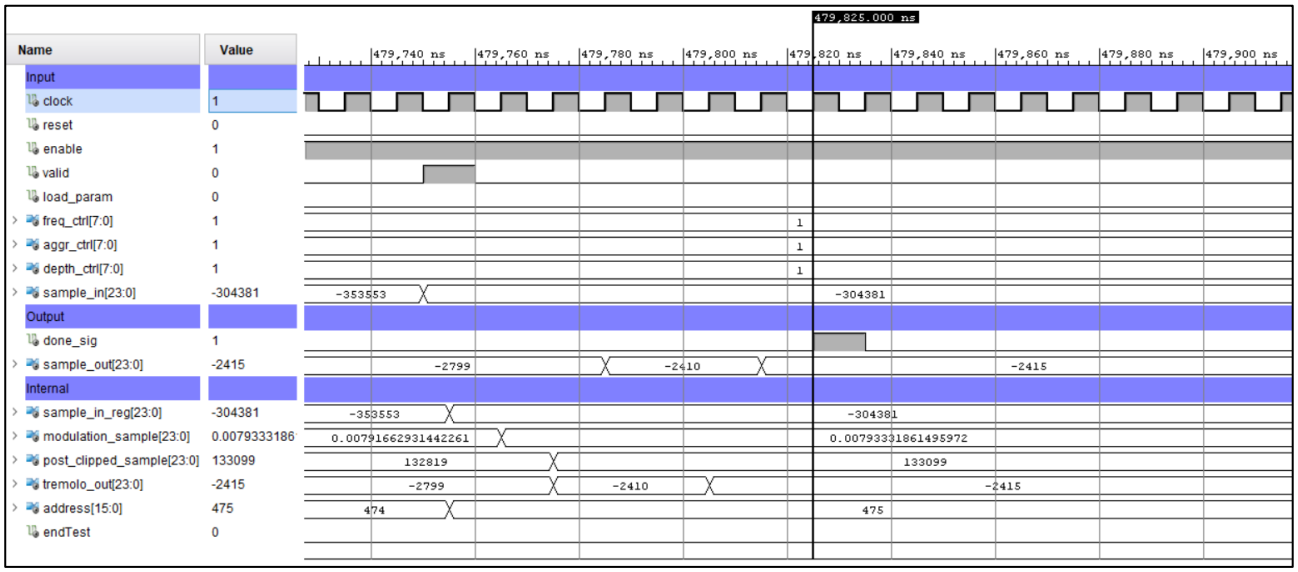

Figure 16 - Done signal

Figure 16 shows the *done_sig* being held high for one clock cycle, after the output of the effect has stabilised; the signal waits for one clock cycle follow output stabilisation, to reduce timing constraints.

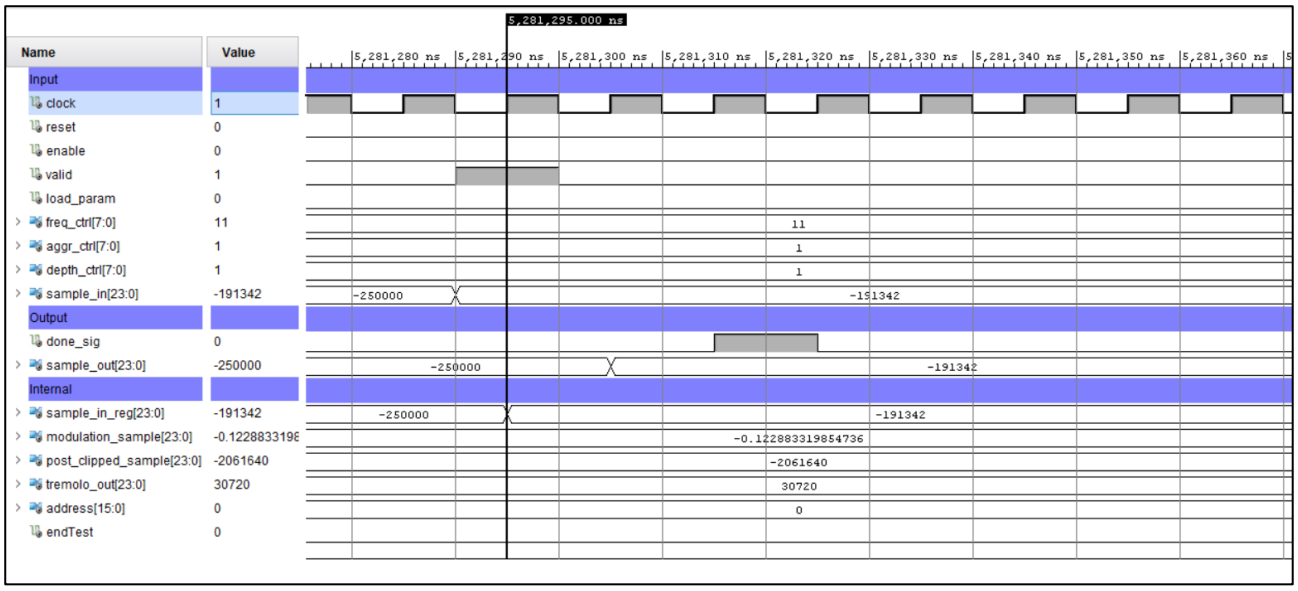

Figure 17 - Enable signal

In Figure 17 it can be seen that the output receives the same value as the input on the clock cycle after it is read into the input register. One more clock cycle later the done signal is raised to indicate that the output is table. This is the true bypass feature of the effect.

The reader may also notice that the address for the ROM is set to 0 when then effect is disabled; this is the desired behaviour.

12 Grade (pass/fail): Pass

13 Approval signatures

Recoverable Signature

X M. Reynolds

Matt Reynolds
Hardware Manager
Signed by: 645892a3-94a8-42ef-8ccc-0fe12b98ad6a

Recoverable Signature

X Simone Ledda

Simone Ledda
Test Manager
Signed by: afcb6896-9ce1-4821-8979-577ea3a903e5