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: 15/07/2020

4 Test date finish: 16/07/2020

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

6 UUT name: delay_BD_wrapper.vhd

7 Hardware block design screenshot

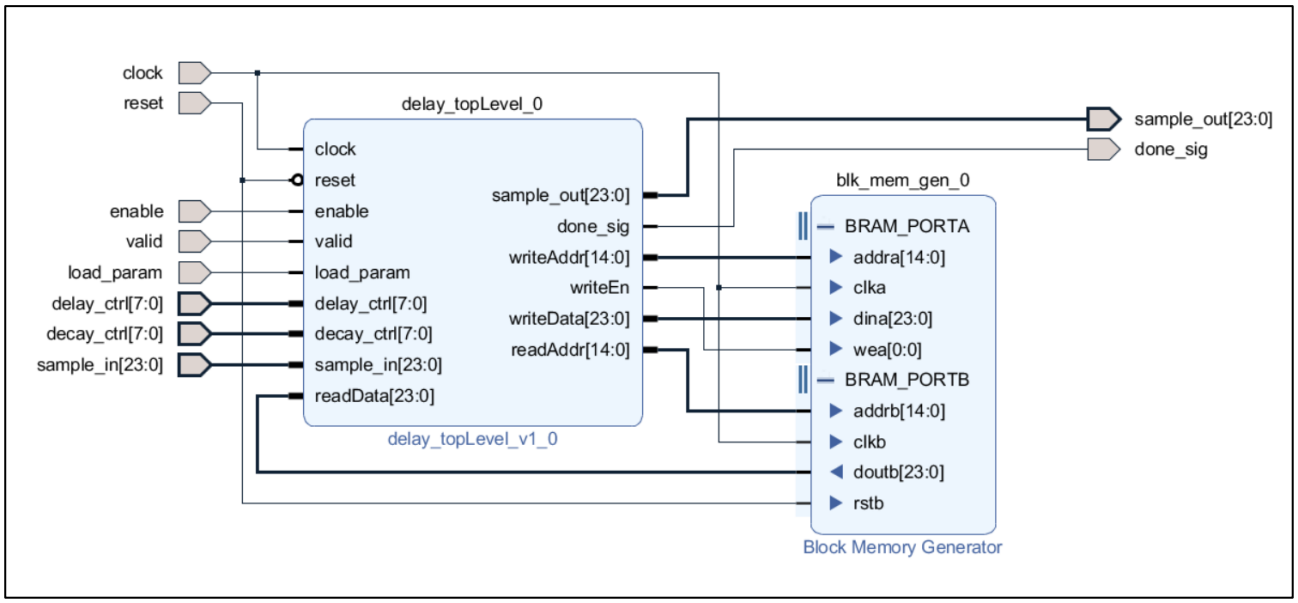

Figure 1 - Top level block design

8 Objective of test

To identify correct:

1. Addressing of the ROM, relative to the delay settings.
2. Division of samples through the feedback loop.
3. Timing of control signals: valid, done_sig, enable.
4. Stabilisation of the output.
5. Operation of the bypass (enable).

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

9 Testbench description / test strategy overview:

The test strategy for this module is focussed on:

1. Validating the objectives (section 8).
2. Performing a hardware-in-the-loop test to confirm audio response.

The test bench uses four main processes. The first loads data into the *sample_in* port of the DUT. These samples are retrieved from a .DAT file containing a 1kHz test tone, made up of 96 samples (two-wave periods). This file is continually read in a loop until the end of the test is signalled by *endTest* being set high.

The second process takes this input data and copies it to an output file. This allows a reference waveform to be captured, which is identical to the input data, but runs for as long as the test does.

The third process passes *sample_out* values to an output file so that the process data can be viewed on an x, y plot against the reference. This is timed in such a way that it is synced with the samples being passed in. This mimics what the CODEC would see and prevents the capture of excessive amounts of data (which is what would happen if data were collected on every clock cycle).

Finally, the fourth process provides the test stimulus. Here the objectives are met. The delay control is increased past its maximum value and the output is checked. When *delay_ctrl* = 1, the output of the delay block should be 1200 samples behind the output of the DUT. This can be seen by the read address of the ROM being behind the write address by 1200 (and checking the outputs).

Following this, the bypass feature is checked by writing *enable = '0'* for two full wave periods. This should result in the output receiving the input and be seen in the x, y plot as unprocessed samples.

The delay and decay controls are then set to minimum values of 1, before looping through increased decay rates. The output of the decay should be the input to the decay entity divided by two to the power of the *decay_ctrl*.

10 Test results

Figure 2 allows the reader to visualise the overall test strategy for this module. You can clearly see the *delay_ctrl* increasing in value for the majority of the test. Each increment increases its wait time relative to the total delay to ensure that the addresses can be checked appropriately and the data out captures at least two full wave periods. The delay range is from 1 to 10. The reader can see that the test increases the *delay_ctrl* to 11, this is to test the incorrect input handling within the circuit.

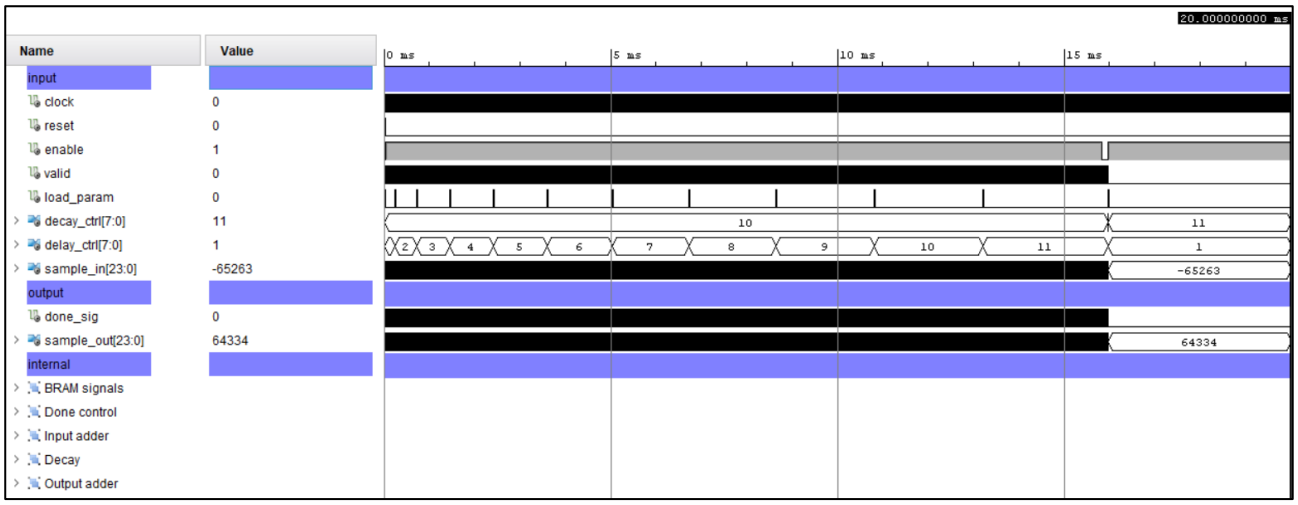

Figure 2 - Testbench overview

The bypass is tested towards the end of the testbench and can be seen dropped low after 15 ms (Figure 2).

Figure 3 shows that the circuit enters the correct state when the reset signal is held high. Setting all registers to zero values.

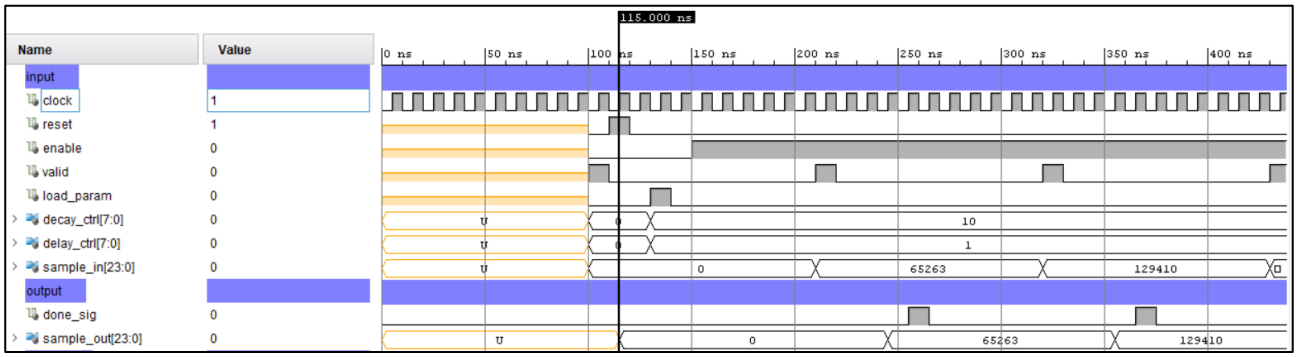

Figure 3 - Reset state

In Figure 4 it can be seen that when enable is set low, the output receives the un-processed input samples. This demonstrates correct operation of the output mux.

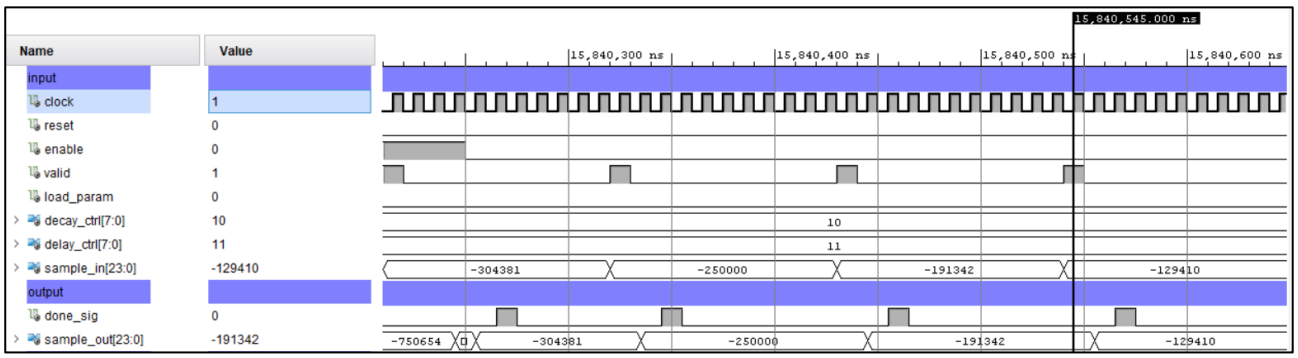

Figure 4 - True bypass (enable)

Figure 5 provides the reader with a clear observation of the RAM addressing used.

Delay = 2. Write address = 3919. Read address = 1518.

This shows correct operation, where the scaled offset is 1200 address locations: 2*1200 = 2400. Plus the read address is one location behind due to the sampling offset. Therefore: 3919 - 1518 = 2401.

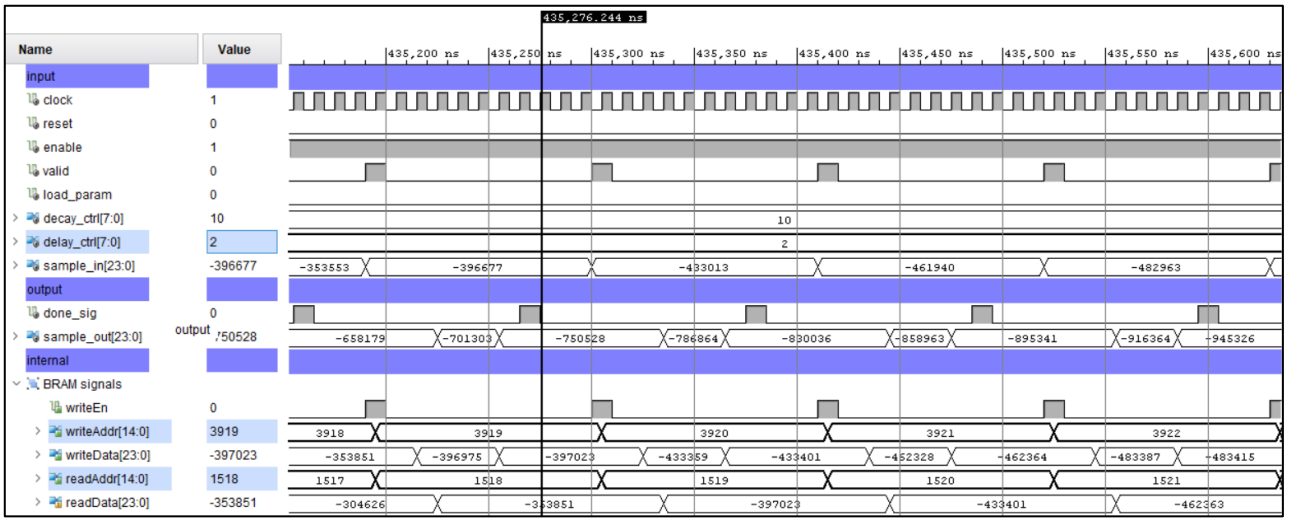

Figure 5 - Write and read address 1

Furthermore, from Figure 6 the reader can observe that the read address control logic handles wrap-around. This is where the offset crosses the zero-boundary of the RAM, hence producing the ring-buffer behaviour.

Delay = 5. Write address = 3409. Read address = 21399.

5*1200 = 6000. 3400 - 6000 = -2600. 24000 - 2600 = 21400.

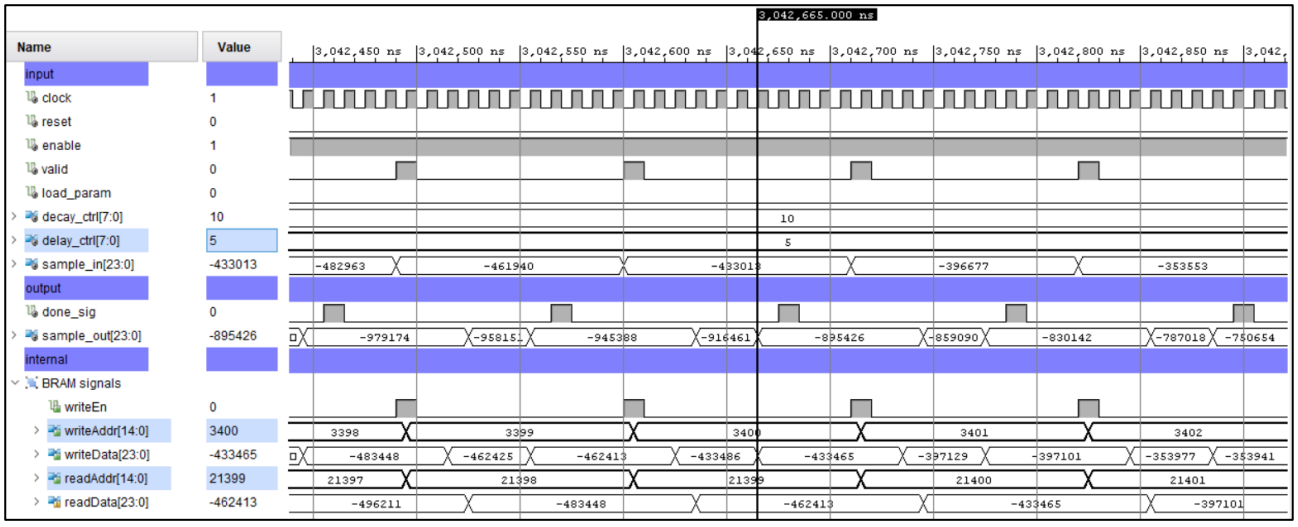

Figure 6 - Write and read address 2

Figure 7 shows correct operation of the decay feed-back loop. Where the decay value corresponds to the divide-by-shift operation:

Decay = 7. Input = 482963. Output = 3773.

482963 / (2^7) = 3773. (rounded)

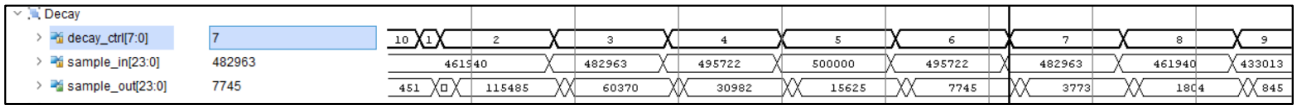

Figure 7 - Decay control

11 Observations:

The GNU plot output data did not prove useful in this test. Further testing will be carried out as hardware-in-the-loop test.

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