ADC-DAC Interface

Anugole Sai Gaurav, 170070008
Anwesh Mohanty, 170070009
Vedant Satav, 170070011
Parth Patil, 170070012

May 2, 2019

1 Aim

1. To implement an interface of an ADC-DAC using a finite-state machine
2. To implement a low pass moving average filter using FSM

2 Experiment setup

The available VHDL files were:
ADCconv.vhdl
filter.vhdl
DUT.vhdl

3 Abstract

The analog-to-digital converter works on the Sampling theorem, which states that a continuous signal can be represented by its samples and can be completely reconstructed when the sampling frequency is atleast twice the frequency of the continuous signal. In out set-up, ADC0804 acts as the analog to digital converter and the DAC0808 as the digital to analog converter. We use the Krypton board to control the pins of the set-up so that the ICs can carry out Analog-Digital-Analog conversion.

4 Algorithm

4.1 Excitation of ADC

Working of the ADC is controlled by exciting its pins at specific times, known as interfacing. The following gives us the interface of the ADC:

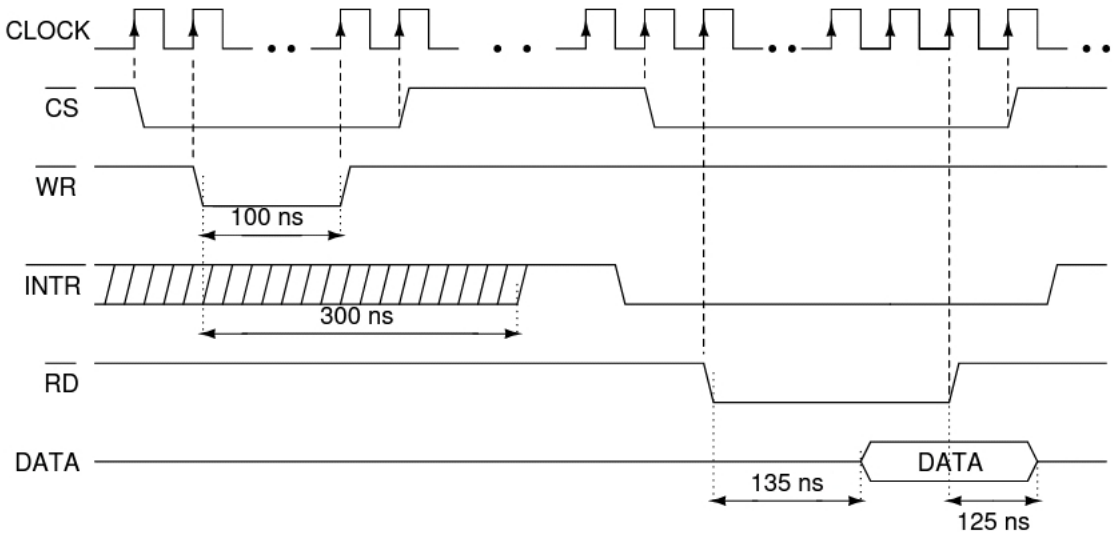


Figure 1: Interface of ADC

The given interface can be broken up into 8 consecutive parts and hence, 8 FSM states:

FSM State	Description
IDLE	The machine starts from this state and is the standard reset state. In this state, CS, WR and RD must all be low. Once a message to begin is received, the machine enters the next state
TRAN1	This is the first transition state in which CS is made high indicating that the IC has to be activated to receive its analog signal. This lasts for 1 clock cycle
SAMP	In this state, the actual sampling of the data takes place. Hence, WR pin is to be made high.
TRAN2	The second transition state involves the WR pin becoming low again to indicate the end of sampling. Now, the ADC processes the received data.
CALC	ADC processes a sample of the analog signal and makes its INTR pin high after it is done. Its time for DAC to read this data.
TRAN3	The third transition again makes the CS pin high, to read the converted data output.
READS	In this state, DAC reads in the the converted data. RD pin is made high. After a sufficient time has elapsed, it presents its output.
TRAN4	The last transition state involves disabling the reading of the DAC and to bring the machine back to its idle state

We maintain two std_logic_vectors to keep a track of the number of cycles that are covered, considering Krypton board uses a clock of 50MHz frequency.

4.2 Low Pass Filter

We use a non-causal low pass filter which averages over the past 8 values of the signal,ie.:

$$y(k) = \frac{\sum_0^8 x(k-m)}{8}$$

This is known as the moving average filter. Each of the x's is 8-bit std_logic_vector type. We maintain the values of the last 7 inputs to the filter in registers in order to calculate the present output. Once an input is received, the values of these registers are updated.

5 Observations

5.1 Part 1

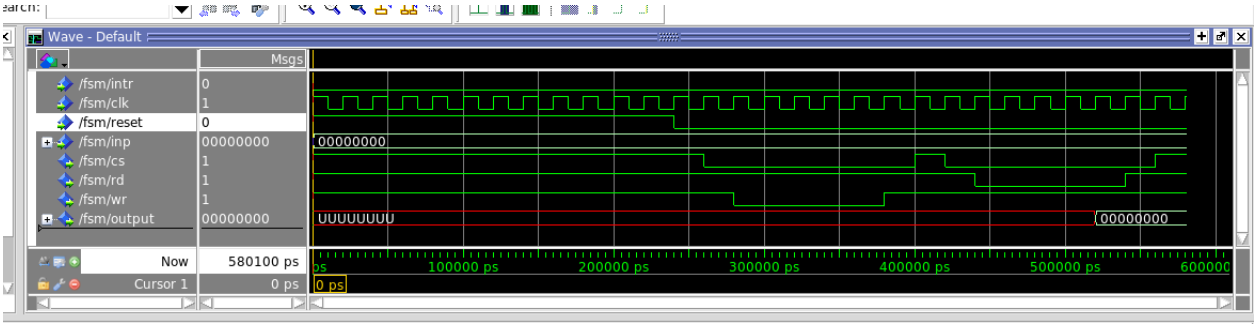
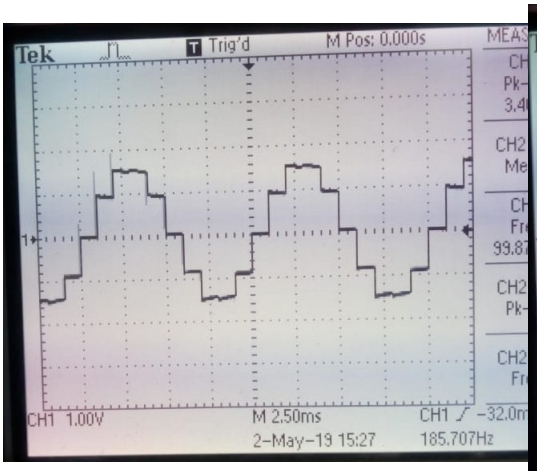
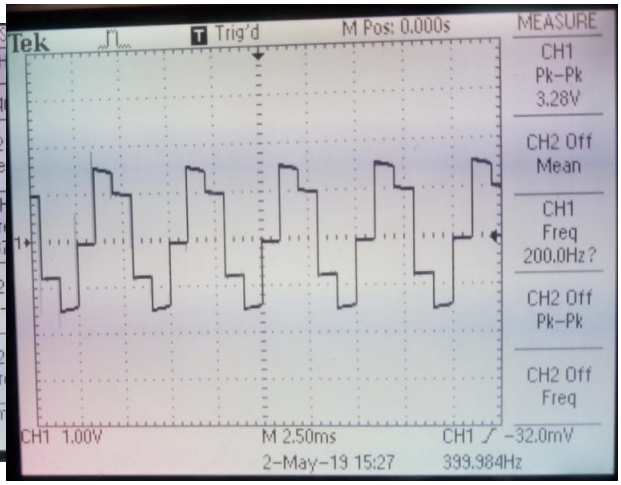


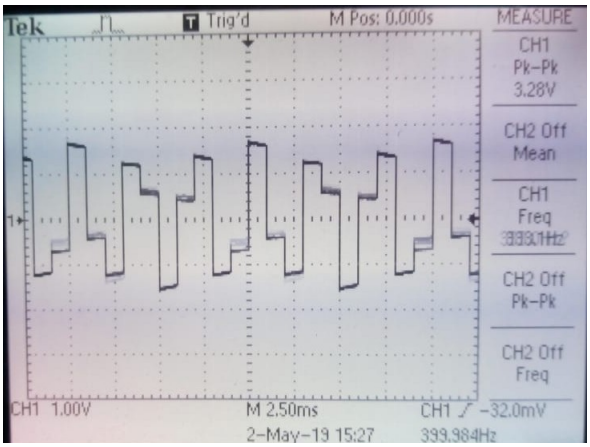
Figure 2: Forced Simulation



(a) F = 100Hz

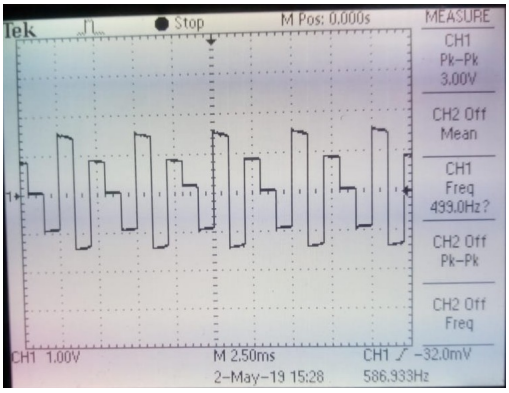


(b) F = 200Hz

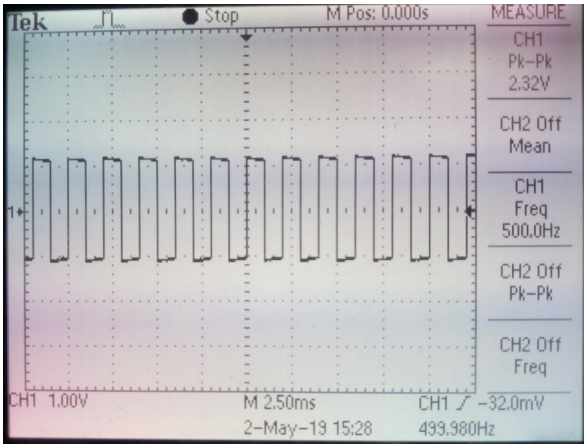


(c) F = 300Hz

Figure 3



(a) F = 400Hz



(b) F = 500Hz

Figure 4

5.2 Part 2

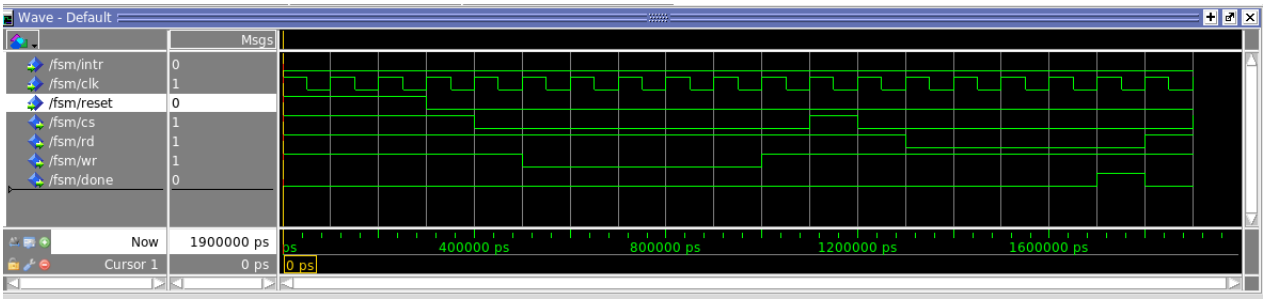


Figure 5: Forced Simulation of ADC conversion

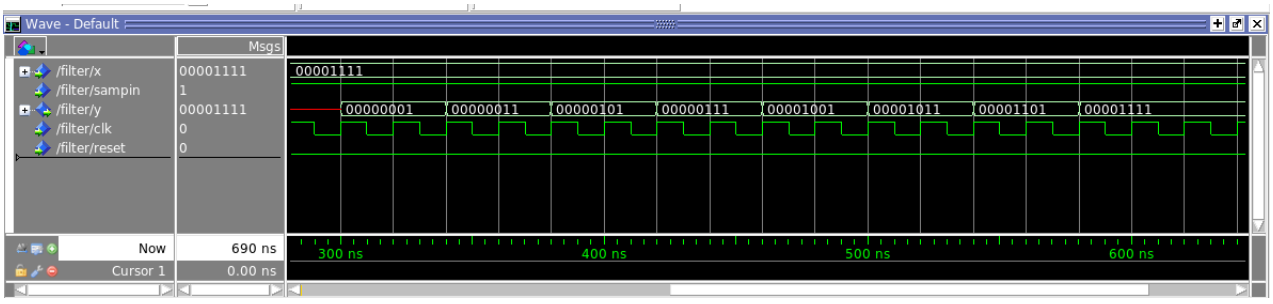


Figure 6: Forced Simulation of digital filter

Vin = 1.5V

Vout	Frequency
2.44	50
0.8	100
0.04	125
0.56	150
0.64	200
0.04	250
0.56	300
0.36	350
0.04	375
0.36	400
0.48	450
0.04	500

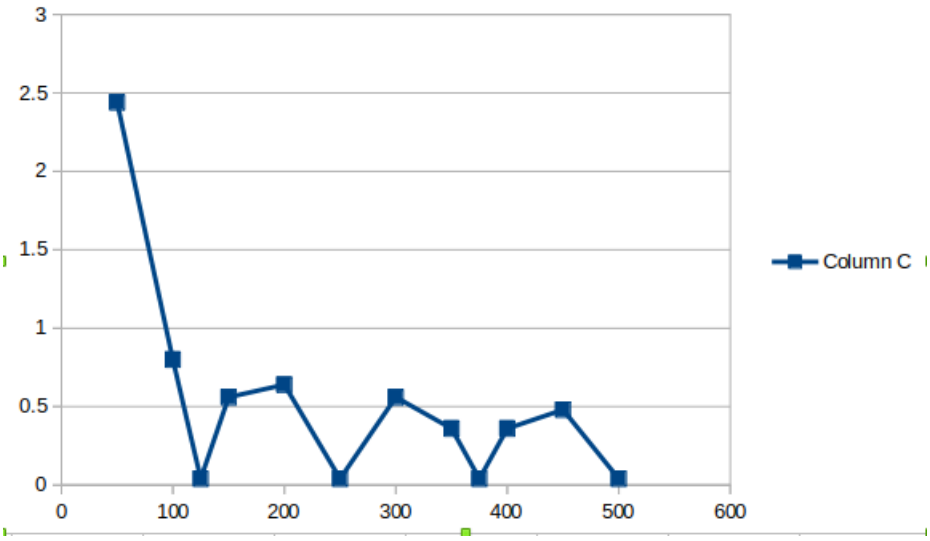
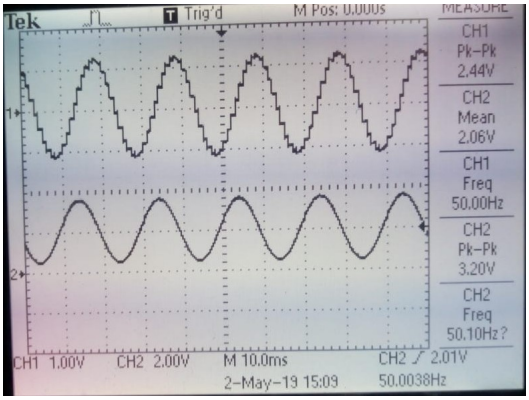


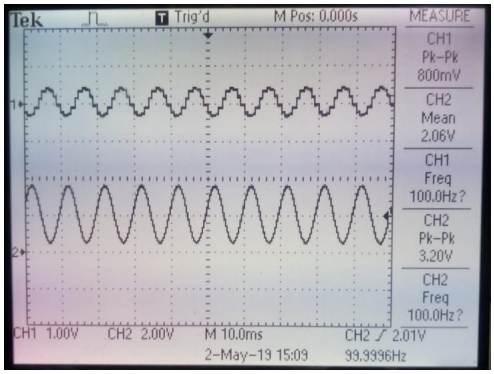
Figure 7: Frequency response

The trasfer fuction is given by:

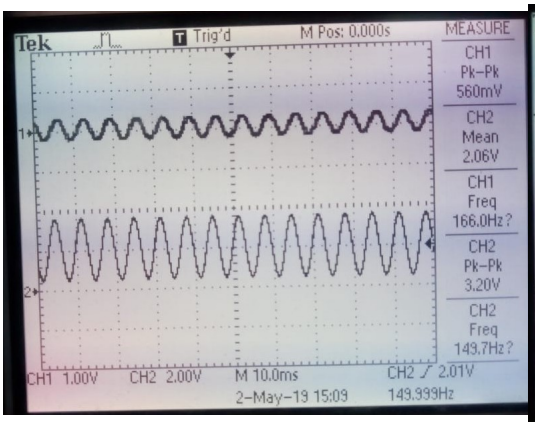
$$H_d(\omega) = \frac{1 - e^{-8j\omega}}{1 - e^{-j\omega}}$$



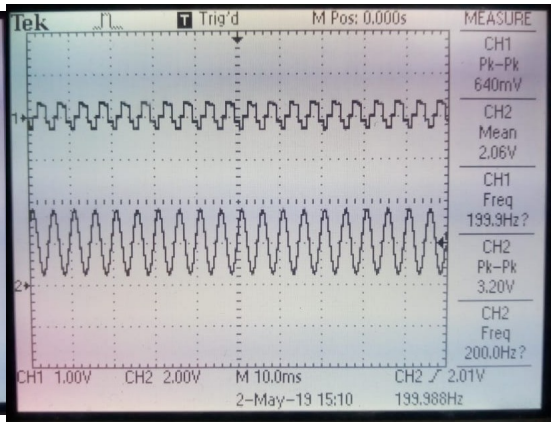
(a) F = 50Hz



(b) F = 100Hz

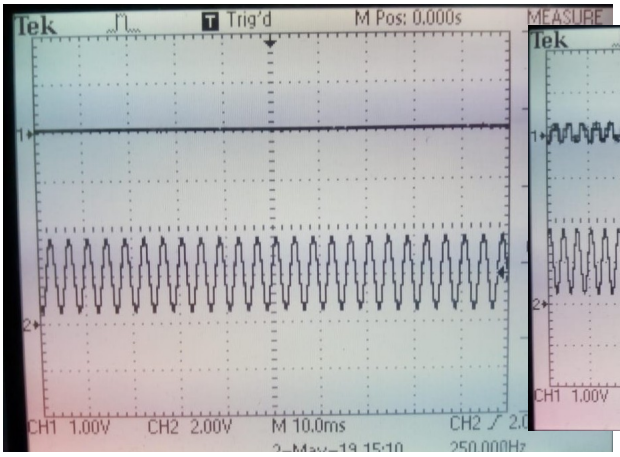


(c) F = 150Hz

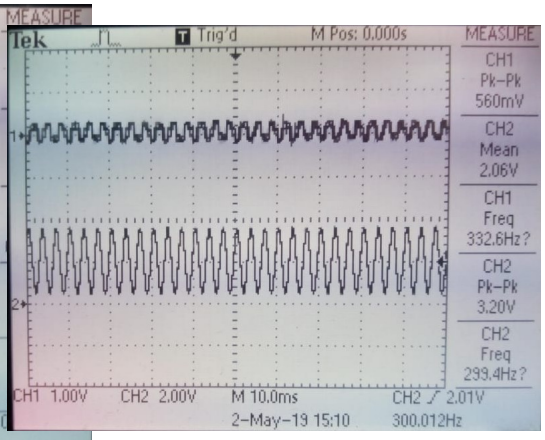


(d) F = 200Hz

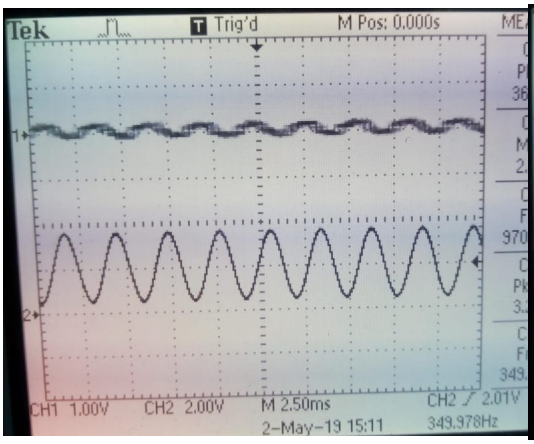
Figure 8



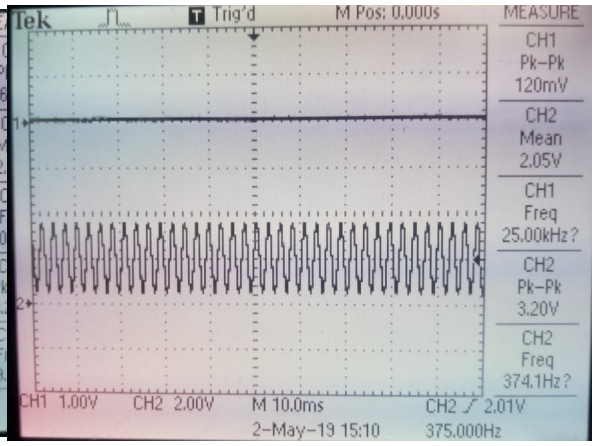
(a) F = 250Hz



(b) F = 300Hz

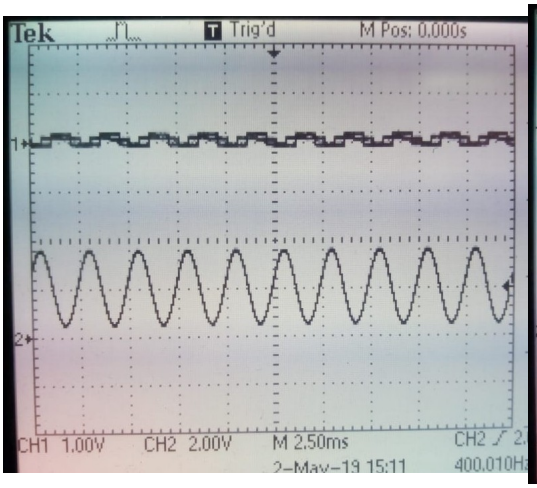


(c) F = 350Hz

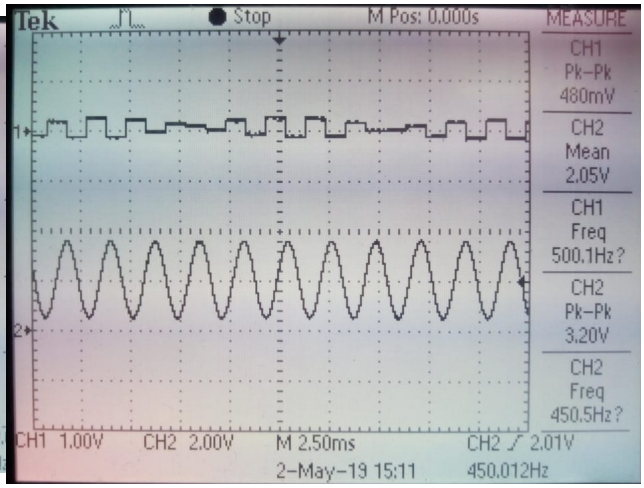


(d) F = 375Hz

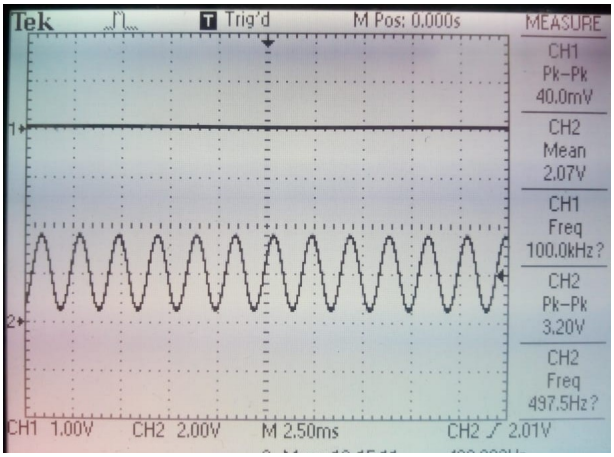
Figure 9



(a) F = 400Hz



(b) F = 450Hz



(c) F = 500Hz

Figure 10

ModelSim report showing the transitions:

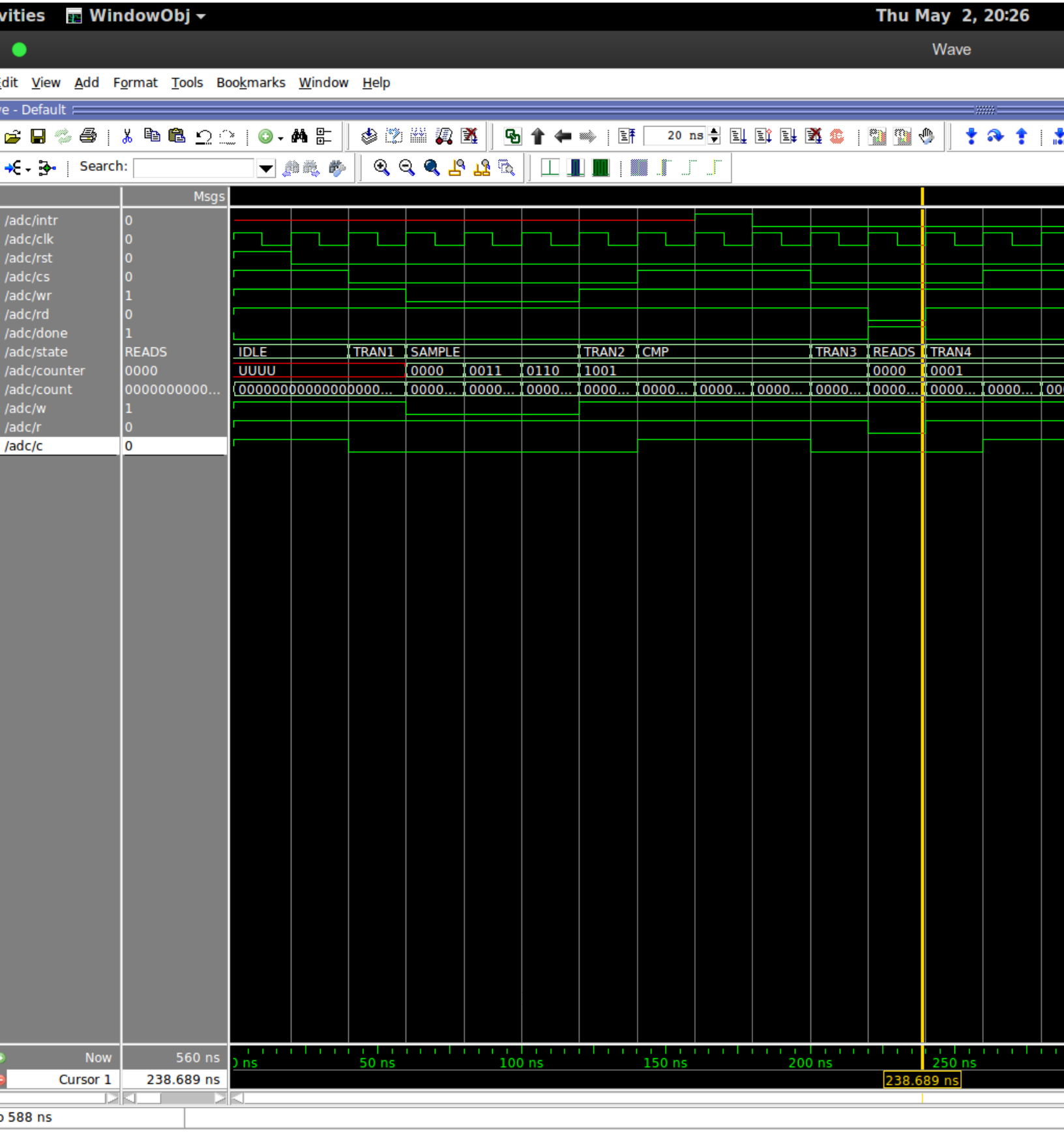


Figure 11: Simulation depicting the interfacing