ADIUVO

ENGINEERING AND TRAINING, LTD.

Mixed Signal Madness

adam@adiuvoengineering.com

Introduction

The world is analog, although it is best processed digitally!

In our programable logic-based solutions we can interface with a wide range of ADC / DAC to convert between analog and digital domains.

Applications which need conversion include wireless communication, RADAR, LIDAR, Signal Processing, Image Processing etc.

Understanding how to work effectively with ADC and DACs is important.

Of course, we will have a lot of fun while doing so and keep our developments targeting cost effective devices for the demo!

Objective

The objective of this session are:

- Outline common Mixed Signal Applications
- Outline basics of ADC and DAC
- Examine Nyquist and the rules of sampling a signal
- Understand Aliasing signals
- Learn how we can work with XADC / Sysmon with external signals
- Creating a Delta Sigma DAC
- Creating a PWM DAC

Typical Applications ADC

- Audio or voice signals: Microphones convert audio/acoustic energy to voltage
- Physical stress: Strain gauges determines the amount of strain when a stress is applied
- Temperature: Thermocouples convert thermal energy to voltage
- Voltage, current: Can be scaled and fed into the ADC as is done in voltmeter/multimeters
- Radio band energy: Amplified and I/Q mixed and down-converted for digital demodulation

Typical Applications DAC

- Audio play back – Convert signals to acoustic energy via speakers
- Telecommunications – Wired and Wireless transmission of information
- Instrumentation and control - convert voltage / current to control actuators, drives and motors.
- RADAR – Generation of RF waveforms

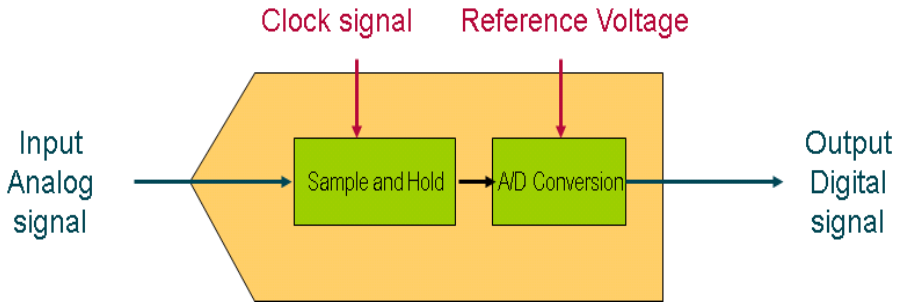
ADIUVO
ENGINEERING AND TRAINING, LTD.

Basics of ADC / DAC

ADC Types

Common methods of implementing Analogue to Digital Convertors are

- Flash convertors are known for there speed and use a series of scaled analogue comparators to compare the input voltage against a reference voltage and use the outputs of these comparators to determine digital code.
- Ramp convertors utilize a free running counter connected to a digital to analogue convertor. The output of the DAC is used to compare against the input voltage, when the two are equal the count is held.
- Successive Approximation convertors are an adaption of ramp convertors and utilize a DAC and a comparator against the analogue input however, instead of counting the SAR converter determines if the analogue representation of the count is above or below the input signal. Allowing a trail and error-based approach to determining the digital code.



ADC Errors

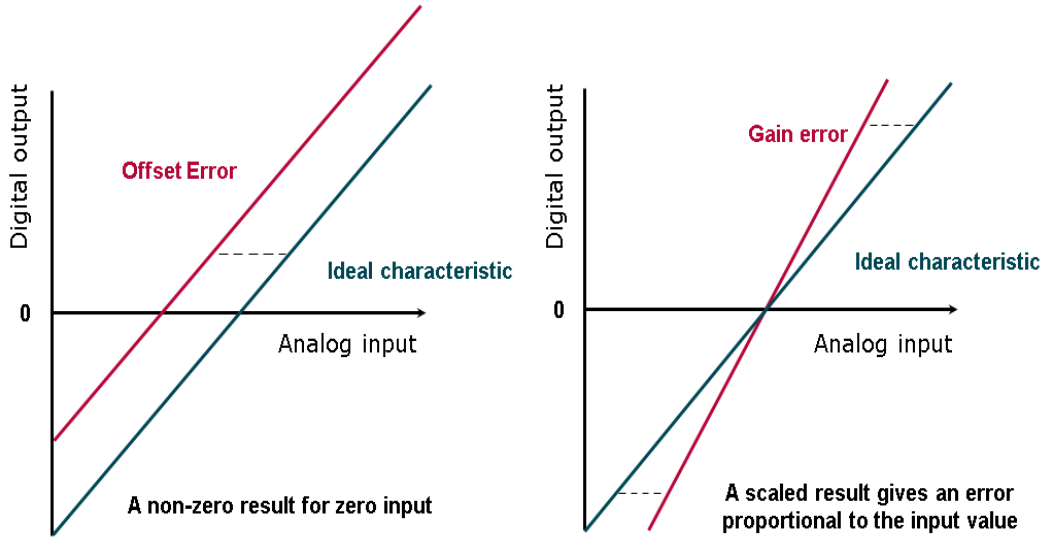
ADC conversion flows Y = mX + C

Y = Digital Output

X = Analog Input

m = number of digital codes which occur for a change in analog input.

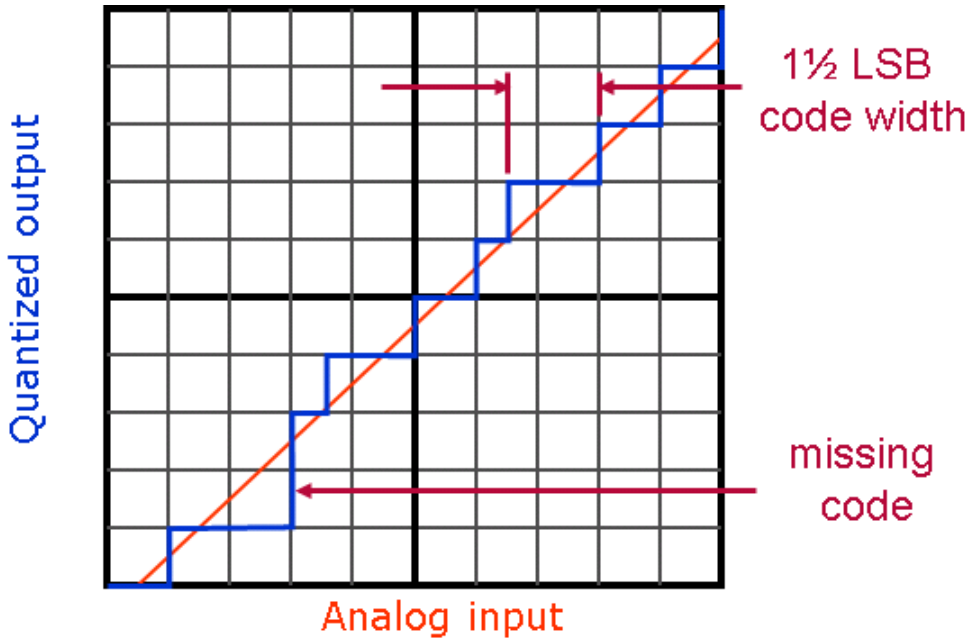
C = minimum output value then the lowest analog voltage is applied



C contributes an offset error, while m provides a gain error

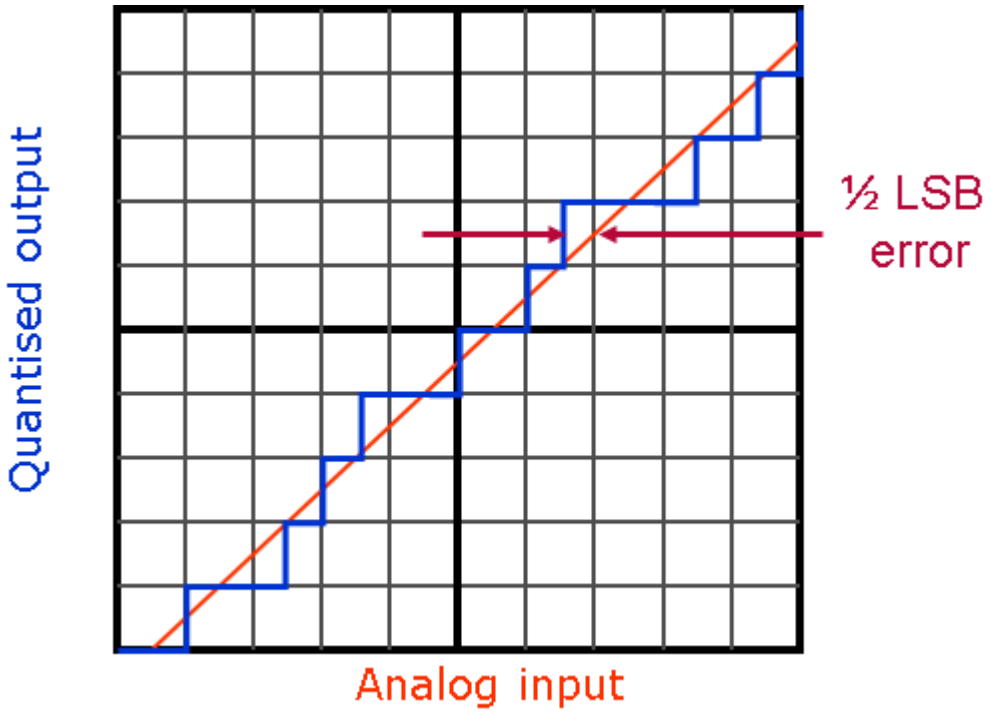
ADC Imperfections

- Differential Non-Linearity (DNL) – deviation of actual code width from the ideal 1 LSB code width.
- Code widths can be either narrower / wider
- Presents additional noise or spurious signals



ADC Imperfections

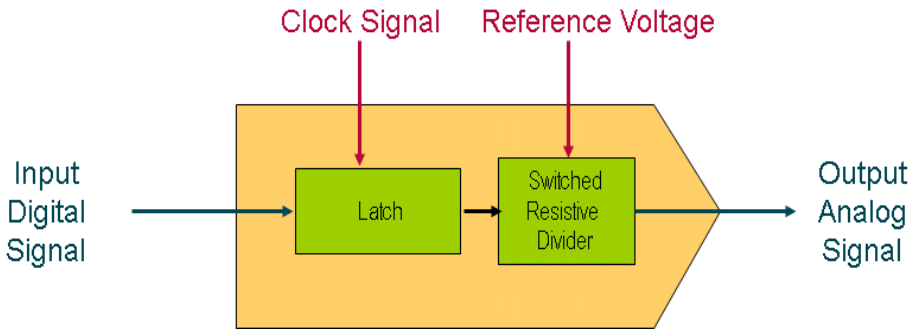
- Integral Non-Linearity – deviation of an actual code transition point from the ideal point on a straight line.
- Calculated after gain and offset errors corrected
- Results in additional harmonics and spurious signals



DAC Types

Common methods of implementing Digital Analogue to Convertors are

- Binary Weighted, these are one of the fastest DAC architectures and sums the result of individual conversions for each logic bit for example a resistor based one will switch on or out resistors depending upon the current code.
- R-2R Ladder, these convertors use a structure of cascaded resistors of value R- 2R. Due to the ease with which precision resistors can be produced and matched these are more accurate than the binary weighted approach.
- Pulse Width Modulation, the simplest type of DAC which passes the PWM waveform through a simple low pass analogue filter. This is commonly used in motor control but also forms the basis for delta sigma convertors.
- Delta Sigma, one-bit DAC that requires only a simple RC filter on the output to recover the signal. To achieve this, the sampling rate of the signal is much higher than the input signal. The delta-sigma ensures the average output level represents the average input level, generating a pulse proportional modulation (PPM) output signal



DAC Errors

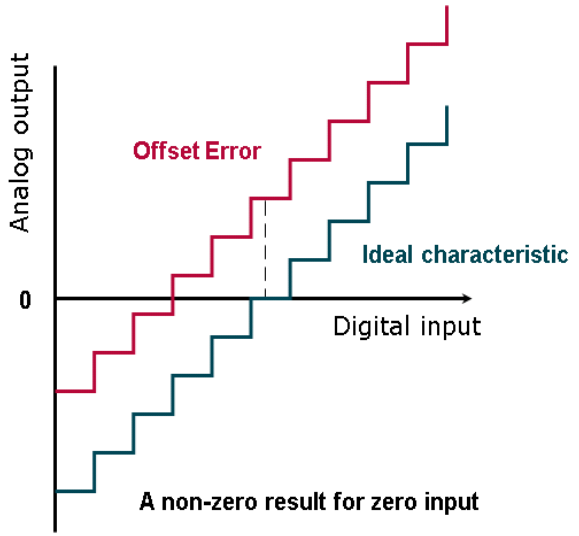
DAC conversion flows Y = mX + C like ADC

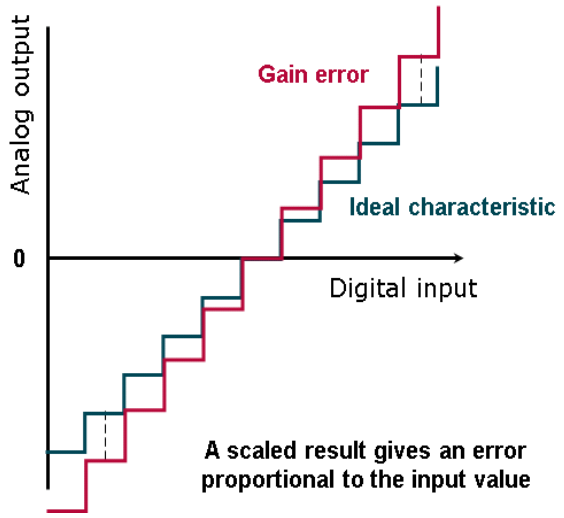
Y = Analog Output

X = Digital Input

m = The range of analoge voltage which occurs for a change in code.

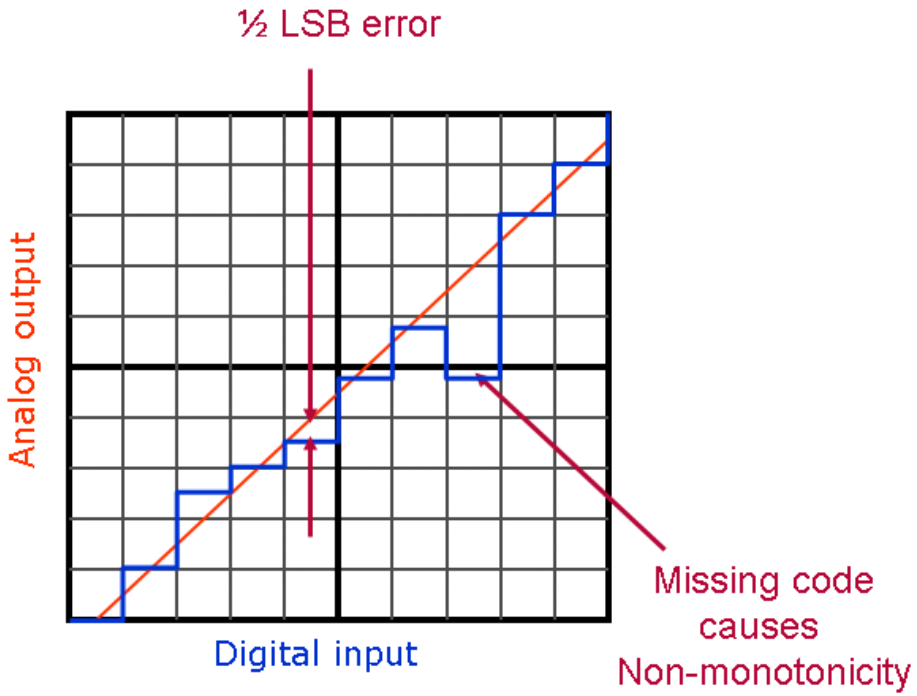
C = minimum output voltage when the lowest code is applied





DAC Imperfections

- Differential non-linearity (DNL) is the deviation of an actual step size from the ideal 1 LSB step
- Integral non-linearity (INL) is the deviation of an actual output voltage from the ideal line drawn between minimum and maximum output levels
- Causes noise/spurs beyond the effects of quantization



DNL = actual - ideal

Key Parameters

Like Selecting a FPGA, we need to select the right ADC / DAC along with sampling frequency and interfacing standards

- Signal to Noise Ratio (SNR) - SNR represents the quantization noise.
- Spurious Free Dynamic Range (SFDR) - the ratio between the input signal and the next highest peak, this is usually a harmonic of the fundamental.
- Effective Number of Bits (ENOB) – the effective number of ADC/ DAC bits determined from the SNR and SFDR.

Calculating Parameters

From testing the ADC / DAC we can determine the SFDR, SNR and ENOB.

Theoretically the SNR can be calculated by the equation

SNR = 6.02N + 1.76 dB Where N is the resolution.

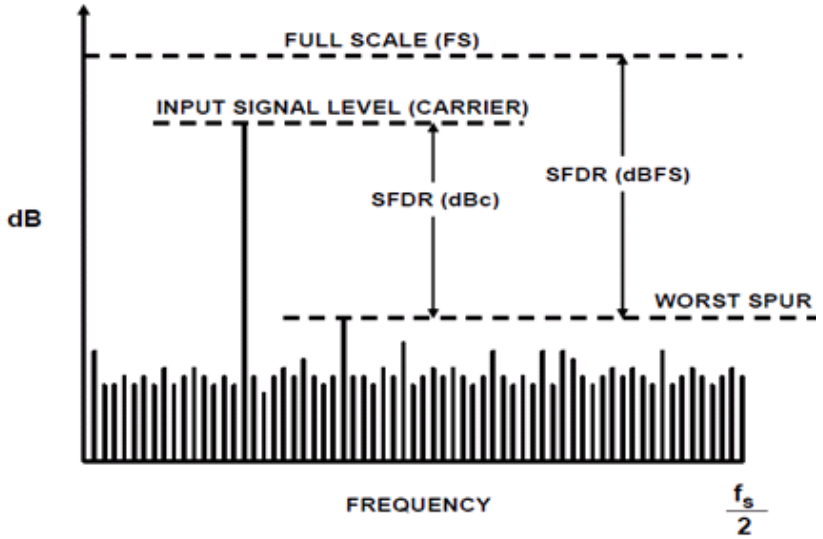
The actual SNR can be determined during system test by taking a FFT of the output and measuring between the value of the input signal and the noise floor

Calculating Parameters

The SFDR is the ratio between the input signal and the next highest peak, this is usually a harmonic of the fundamental normally the SFDR is given in terms of dBc and will degrade as the input signal power reduces.

Using the measured SNR, we can determine the ENOB

ENOB = (SNR – 1.77 / 6.02)



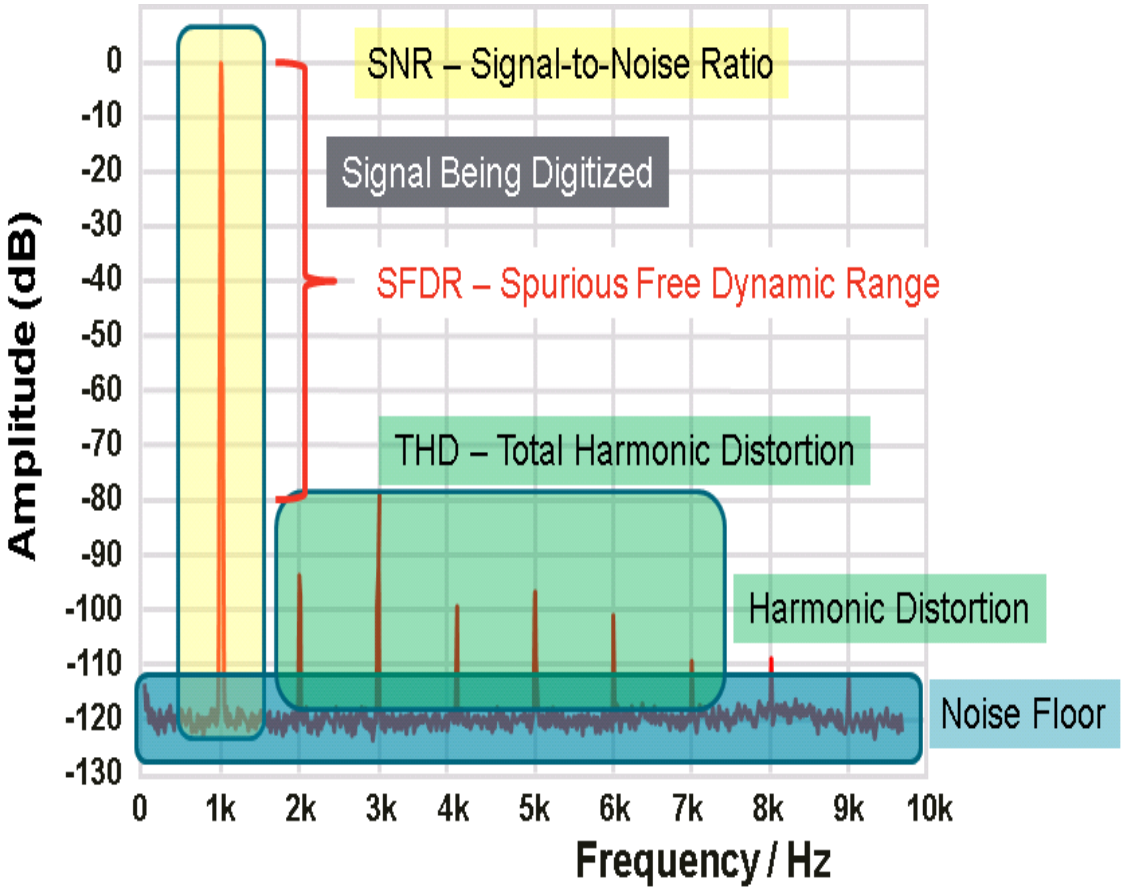
FFT Considerations

To make accurate measurements we need to ensure the FFT is correctly sized.

This ensures the FFT noise floor is correct

The FFT noise floor is given by

FFT Noise Floor = 6.02N + 1.76 dB + 10 LOG10(FFT Size / 2)



Frequency Spectrum

To work effectively with digital processing, we need to convert from continuous signalling to discrete samples.

Crucially we want to do this without losing information

Nyquist-Shannon Sampling Theorem defines how we can achieve this.

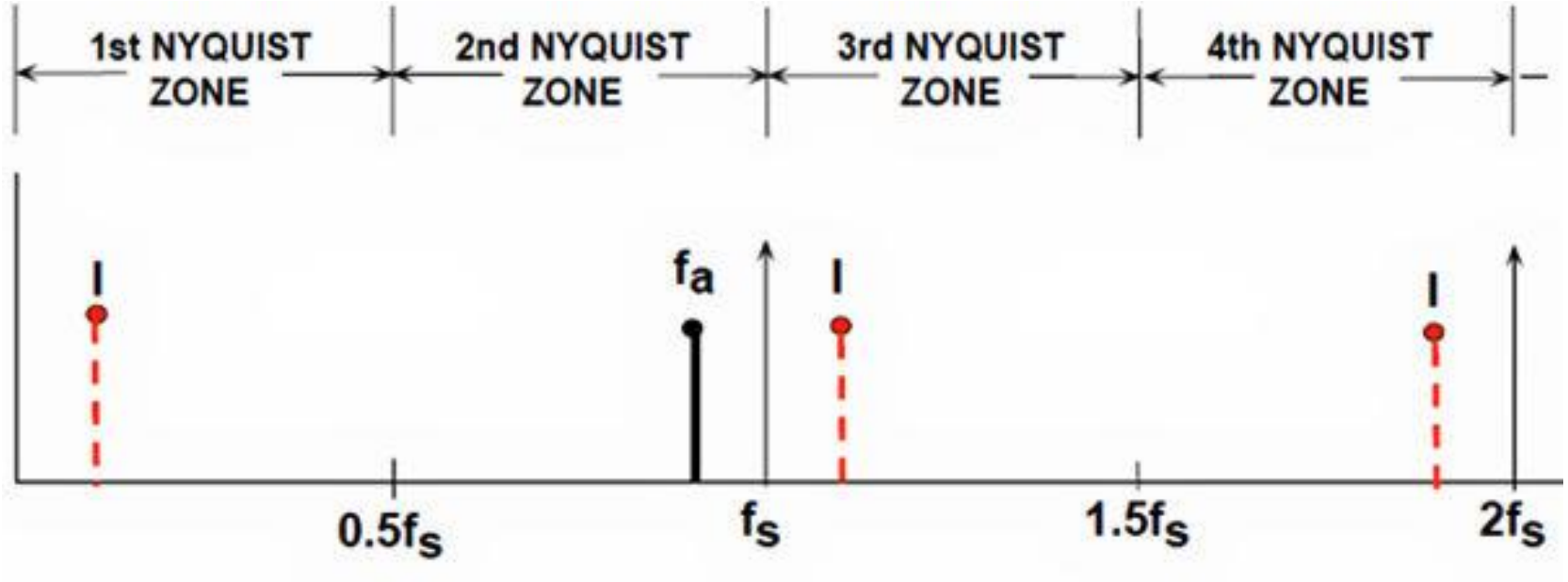
“A continuous signal can be completely represented in its samples and perfectly reconstructed if it is sampled at a frequency that is at least twice the highest frequency component of the signal”

Aliasing

When we below Nyquist limit, we get aliasing – depending on what we are sampling this may or may not be a good thing.

Generally, it can be very useful if we want to use a direct sampling approach.

Selected device needs to have a corresponding input bandwidth to support.



XADC / Sysmon

ADIUVO
ENGINEERING AND TRAINING, LTD.

XADC / Sysmon

Mixed signal device enabled in all Seven Series, UltraScale and UltraScale+ devices

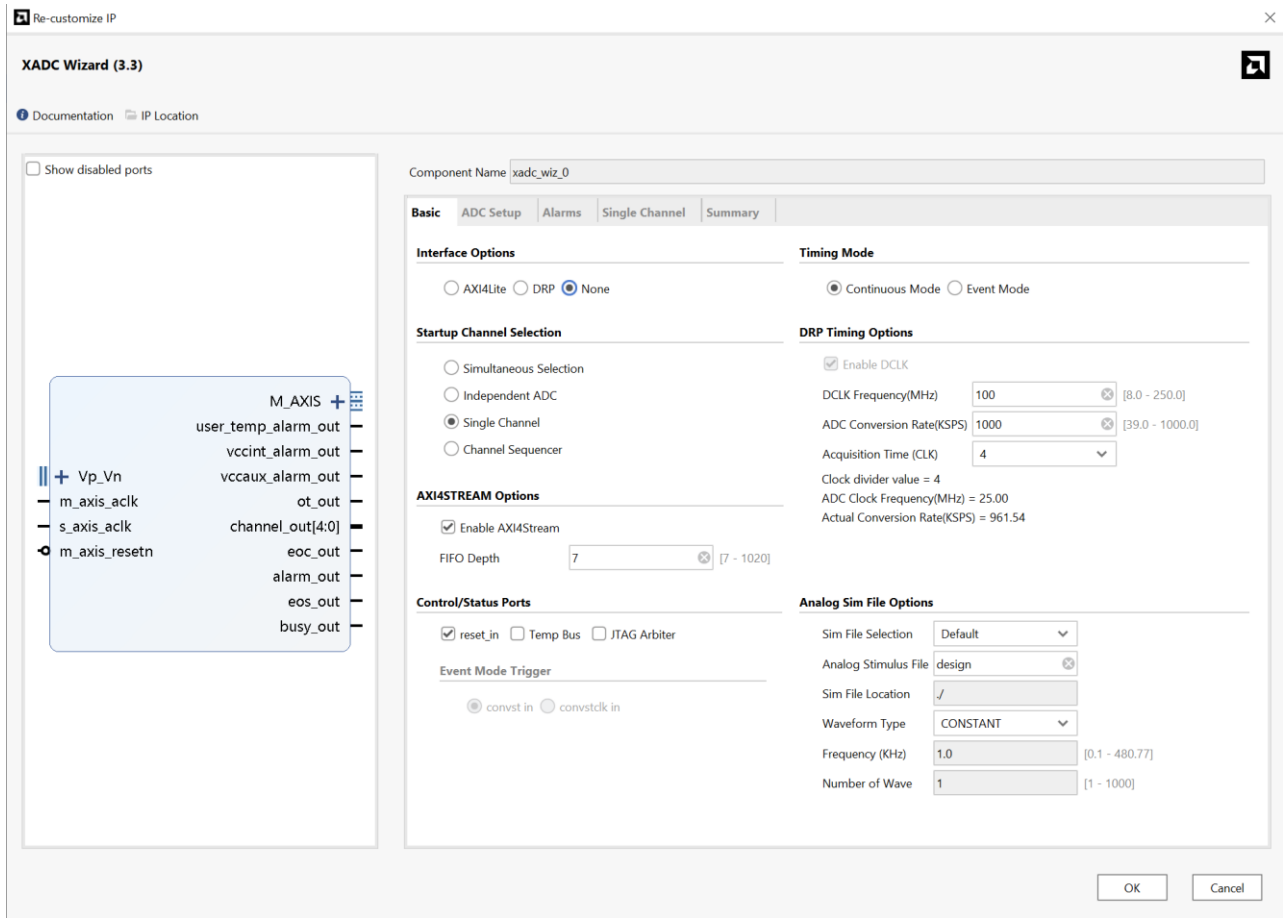
- 1 MSPS (XADC), 200 KSPS (SYSMON)
- 12-bit (XADC), 10-bit (Sysmon)
- Uni or Bi-Polar inputs configuration
- Ability to monitor device internal supply voltages and temperature
- Ability to monitor 17 External signals
- Support for external multiplexer sequencing
- Internal sequencer
- Signal averaging
- Automatic Gain / Offset correction

XADC / Sysmon

Implementation in PL enables configuration using the appropriate wizard.

Ability to directly access the XADC / Sysmon output in the PL

AXI or Vector output depending on XADC / Sysmon



XADC / Sysmon

Very useful for Built in Test and Security monitoring. Enables us to ensure we have everything operating within the correct operating conditions

Within our design we can access these values and set limits, both upper and lower limits on voltages / temperature.

If a level is exceeded the interrupt is generated and processor or PL can address the issue

Can be used to provide an anti-tamper response also

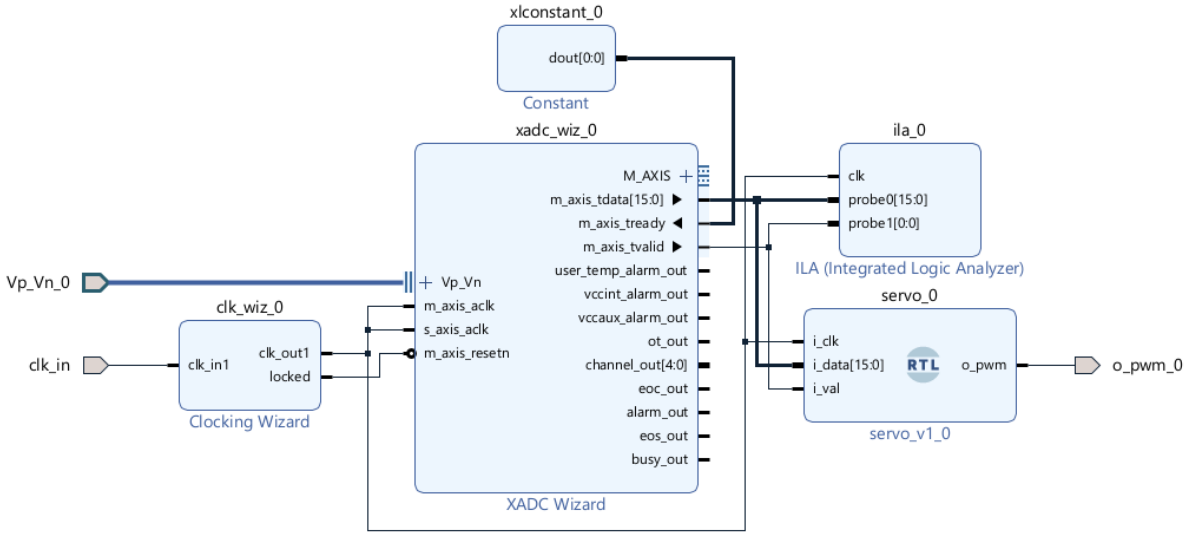
Working with XADC

Leverage XADC or System Management Wizard.

Both are very similar let's look at XADC.

Application to monitor Light Dependent Resistor and and drive servo

Yes it plays the google dino game!



ADIUVO
ENGINEERING AND TRAINING, LTD.

XADC Demo

ADIUVO
ENGINEERING AND TRAINING, LTD.

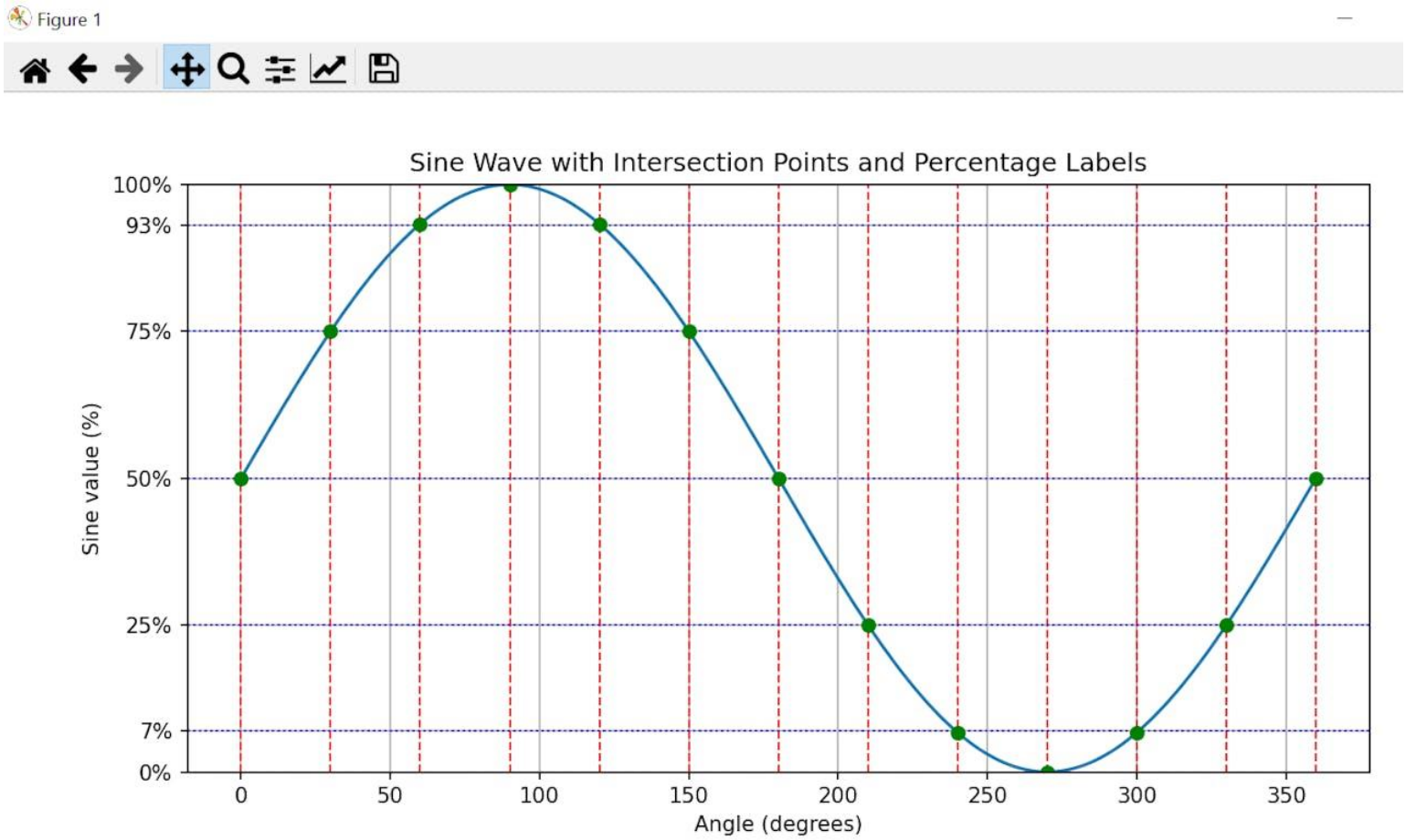
Pulse Width Modulation DAC

PWM DAC

- The simplest of the two schemes is to use a PWM signal to generate the analogue signal.
- We can vary the duty cycle to generate an analogue signal for recovery by low pass filter
- A PWM signal has two key elements
  - Period - The repetition period of the signal
  - Duty Cycle - How much of period the signal is high

PWM DAC

- Create a sine wave using the PWM we need to vary the PWM duty cycle in accordance with Sine wave amplitude.
- Segment into 12 separate segments each which covers 30 degrees and measure the amplitude.



PWM DAC

- If we want to generate a sine wave at 1KHz, generate the PWM output waveform at least 12 times faster than that if we have segmented the waveform into 12 elements.

- This means our PWM frequency needs to be 12000 Hz or have a period of 83.333 Microseconds.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity pwm_sine is port(
i_clk : in std_logic;
o_pwm : out std_logic
);
end entity;


architecture rtl of pwm_sine is

constant upper_width : integer := 14;
constant max_count : integer := 8333;

type t_sine_array is array (0 to 11) of unsigned(upper_width-1 downto 0);
constant analog_wave : t_sine_array := (
to_unsigned(6250 ,upper_width),
to_unsigned(7750 ,upper_width),
to_unsigned(8333 ,upper_width),
to_unsigned(7750 ,upper_width),
to_unsigned(6250 ,upper_width),
to_unsigned(4166 ,upper_width),
to_unsigned(2083 ,upper_width),
to_unsigned(583 ,upper_width),
to_unsigned(0 ,upper_width),
to_unsigned(583 ,upper_width),
to_unsigned(2083 ,upper_width),
to_unsigned(4166 ,upper_width));


signal s_counter : unsigned(upper_width-1 downto 0):=(others=>'0');
signal s_wave_pos : unsigned(3 downto 0):=(others=>'0');

begin

process(i_clk)
begin
if rising_edge(i_clk) then
if s_counter = max_count - 1 then
s_counter <= (others=>'0');
if s_wave_pos = 11 then
s_wave_pos <= (others => '0');
else
s_wave_pos <= s_wave_pos + 1;
end if;
else
s_counter <= s_counter + 1;
end if;
end if;
end process;

process(i_clk)
begin
if rising_edge(i_clk) then
o_pwm <= '1' when s_counter < analog_wave(to_integer(s_wave_pos)) else '0';
end if;
end process;

end architecture;
```

ADIUVO
ENGINEERING AND TRAINING, LTD.

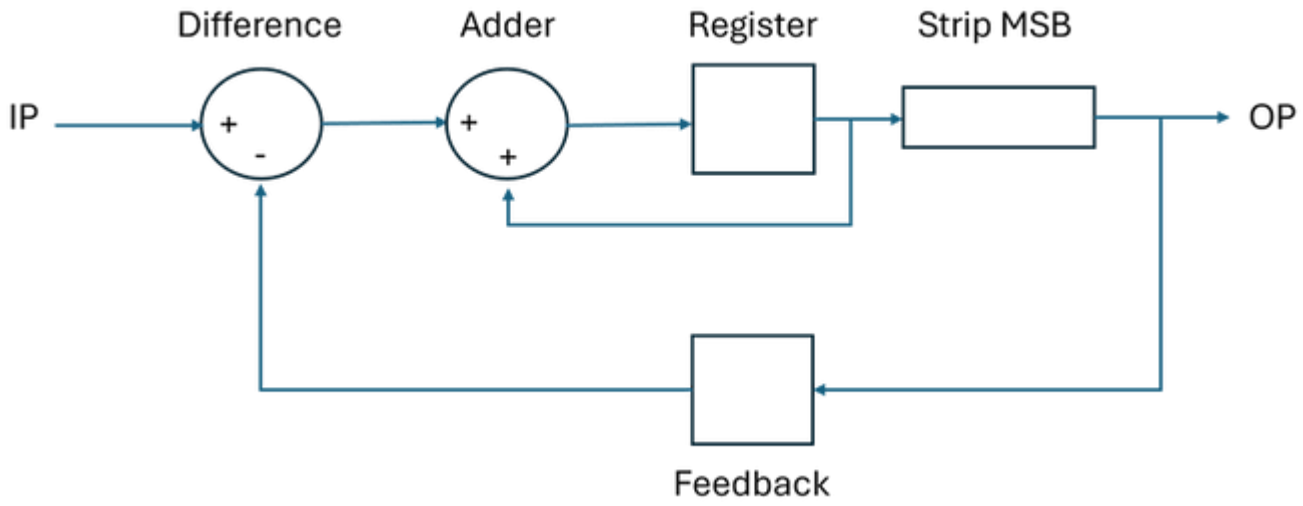
Pulse Width Modulation DAC Demo

ADIUVO
ENGINEERING AND TRAINING, LTD.

Delta Sigma DAC

Delta Sigma

- The delta-sigma works by ensuring the average output level represents the average input level.
- Generates a pulse proportional modulation (PPM) output signal.
- Create 16 bit Delta Sigma



ADIUVO
ENGINEERING AND TRAINING, LTD.

Delta Sigma DAC Demo

Resource Utilization

```
Copyright 1986-2022 Xilinx, Inc. All Rights Reserved. Copyright 2022-2024 Advanced Micro Devices, Inc. All Rights Reserved.
------------------------------------------------------------------------------------------------------------------------------------------------------
| Tool Version : Vivado v.2024.1 (win64) Build 5076996 Wed May 22 18:37:14 MDT 2024
| Date         : Fri Jul 12 13:53:43 2024
| Host         : Laptop-Adam running 64-bit major release  (build 9200)
| Command      : report_utilization -hierarchical -file report.txt
| Design       : delta_sigma_dac_top
| Device       : xc7s6ftgb196-1
| Speed File   : -1
| Design State : Synthesized
------------------------------------------------------------------------------------------------------------------------------------------------------

Utilization Design Information

Table of Contents
-----------------
1. Utilization by Hierarchy

1. Utilization by Hierarchy
---------------------------

+--------------------------+------------------+------------+------------+---------+------+-----+--------+--------+------------+
|         Instance         |      Module      | Total LUTs | Logic LUTs | LUTRAMs | SRLs | FFs | RAMB36 | RAMB18 | DSP Blocks |
+--------------------------+------------------+------------+------------+---------+------+-----+--------+--------+------------+
| delta_sigma_dac_top      |           (top)  |         58 |         58 |       0 |    0 | 104 |      0 |      1 |          0 |
|   (delta_sigma_dac_top)  |           (top)  |         18 |         18 |       0 |    0 |  64 |      0 |      1 |          0 |
|   delta_sigma_dac_inst   | delta_sigma_dac  |         21 |         21 |       0 |    0 |  21 |      0 |      0 |          0 |
|   pwm_inst               |        pwm_sine  |         19 |         19 |       0 |    0 |  19 |      0 |      0 |          0 |
+--------------------------+------------------+------------+------------+---------+------+-----+--------+--------+------------+
```

ADIUVO

ENGINEERING AND TRAINING, LTD.

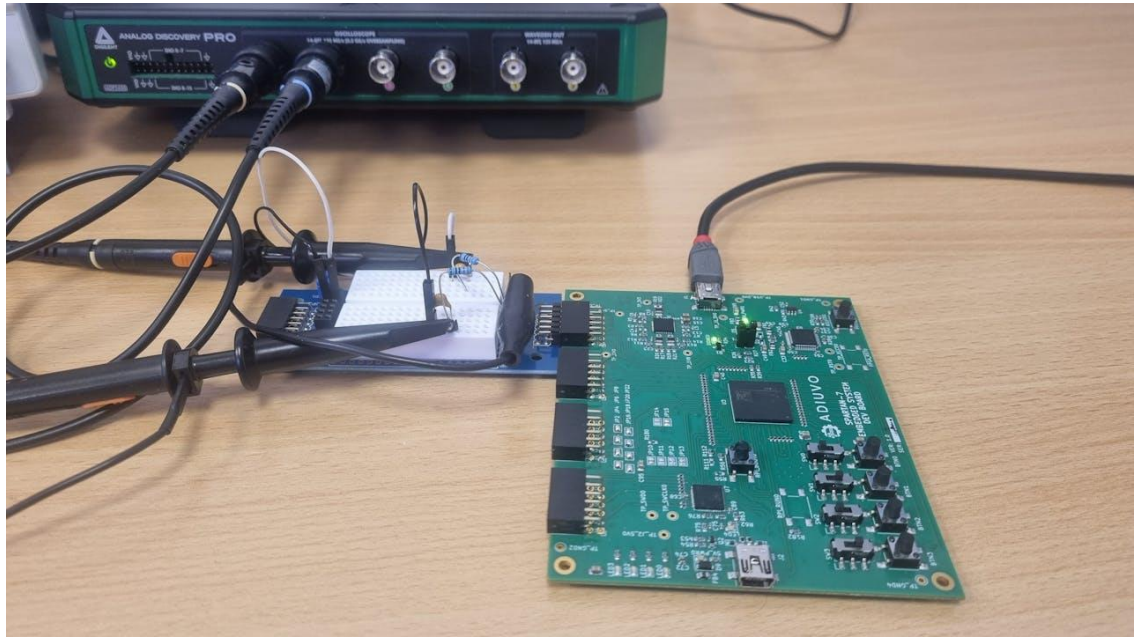
Filter Design

Filter Design

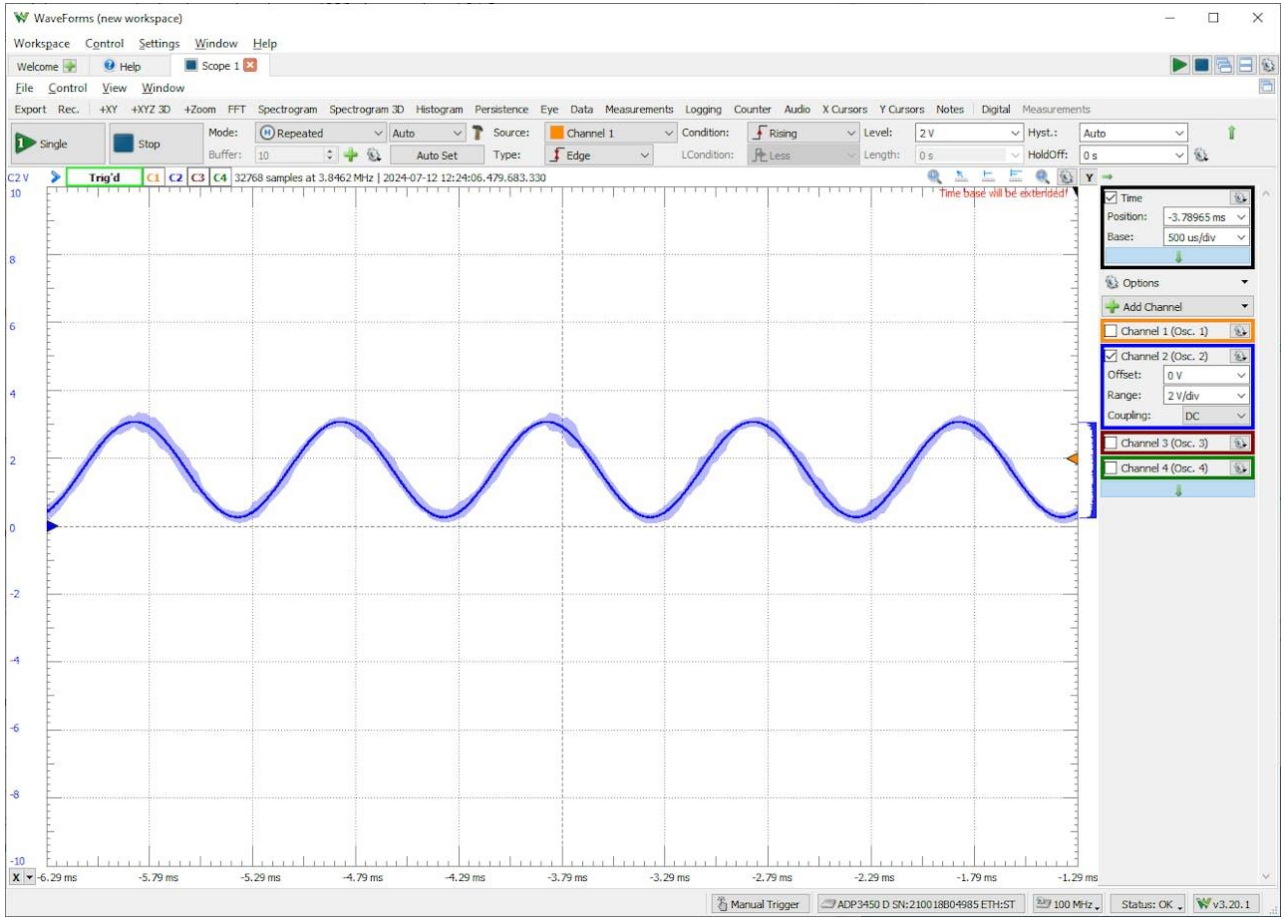
- The easiest way to recover the signal is to use a low pass filter, for this we can use a simple RC filter.
- To create the correct filter, we need to decide on the cut of frequency of Fc. This is the frequency above which we wish to attenuate the signal.
- The equation for a RC filter is as below

$$Fc = 1 / (2\pi RC)$$

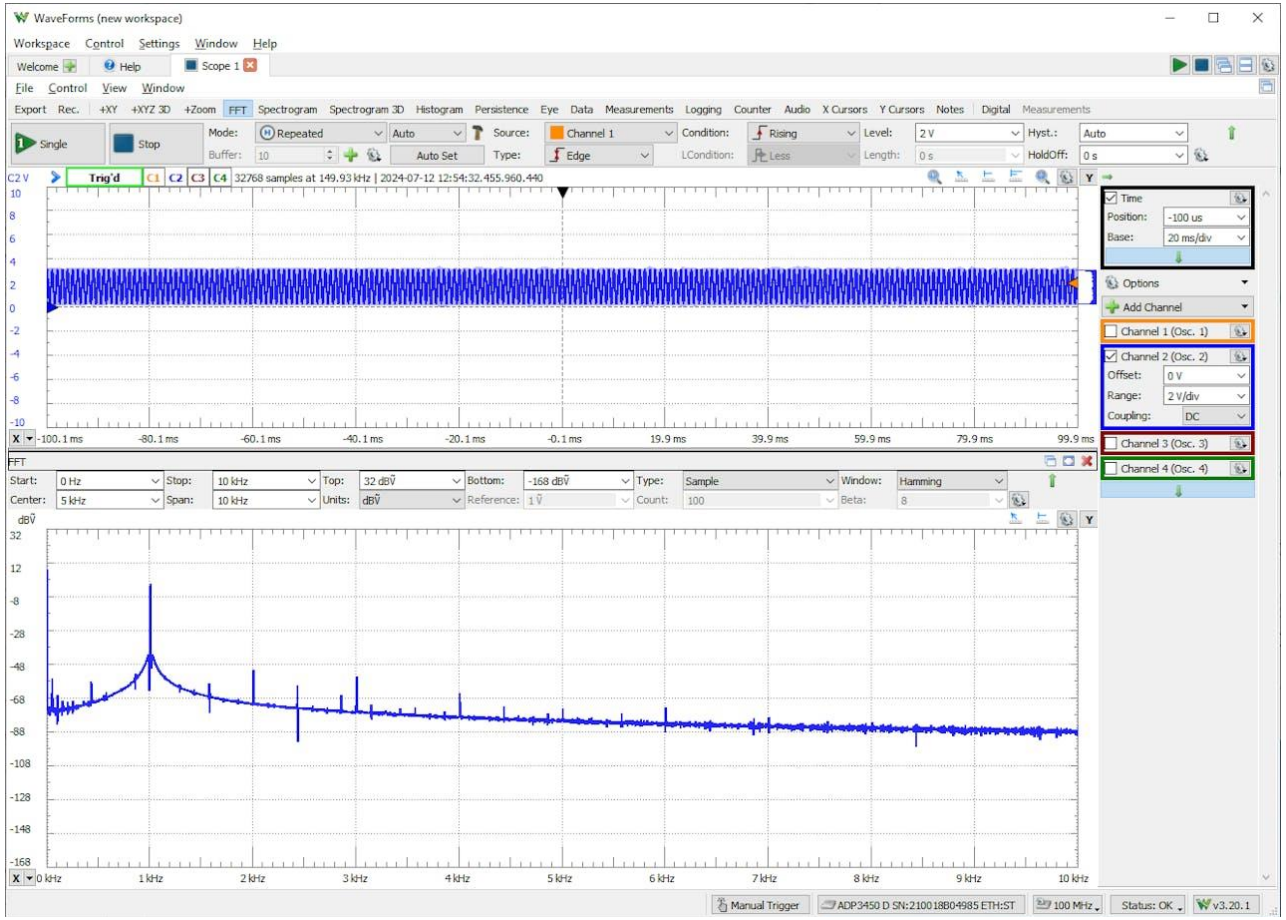
- Fc at 2 KHz and a capacitor value of 100 nF meaning we need a resistor of 820 Ohms.
- With this RC combination we should have a filter which has a Fc of 1940 Hz, which is close enough to the 2KHz cut off.



Delta Sigma Output

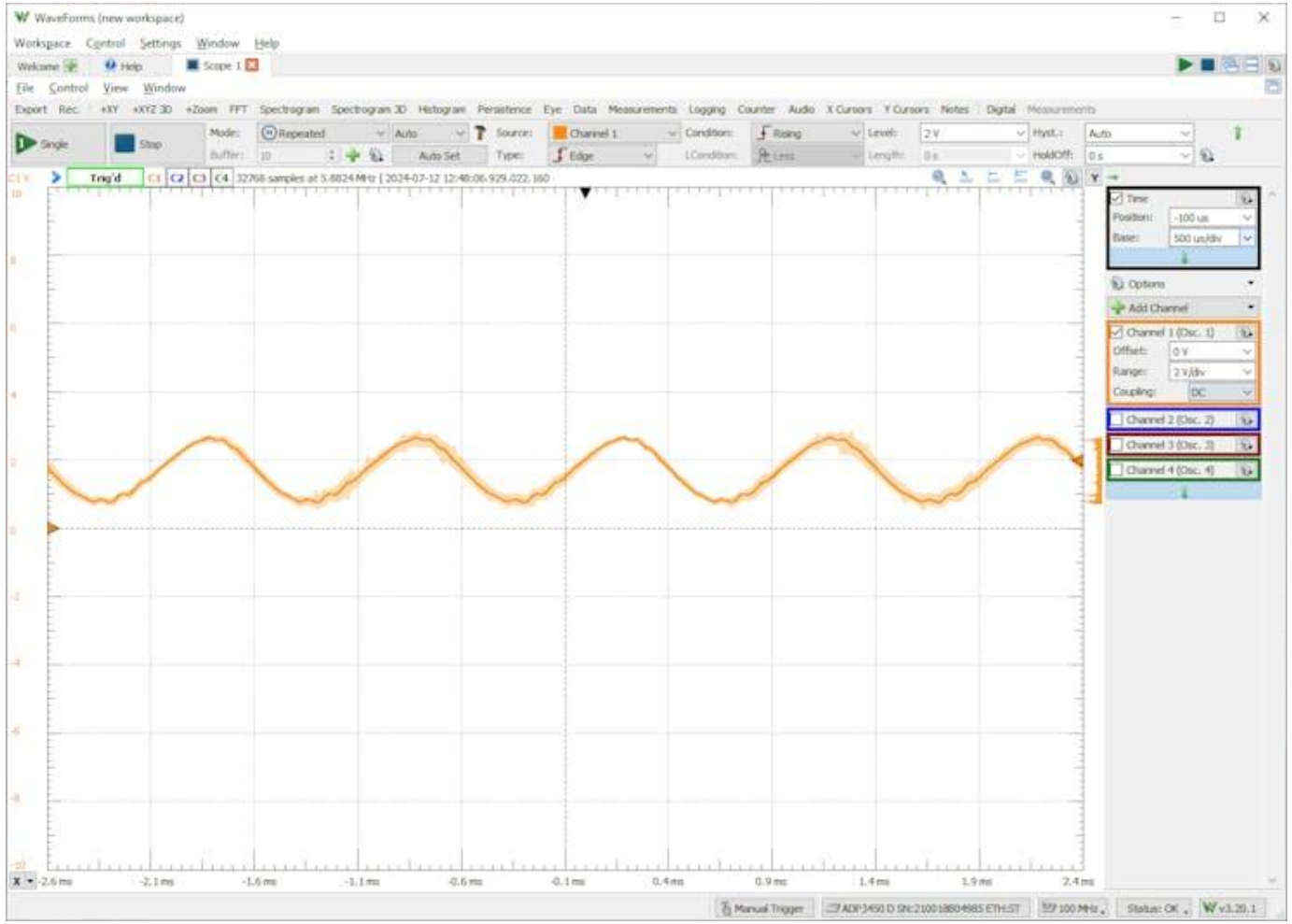


Delta Sigma Output



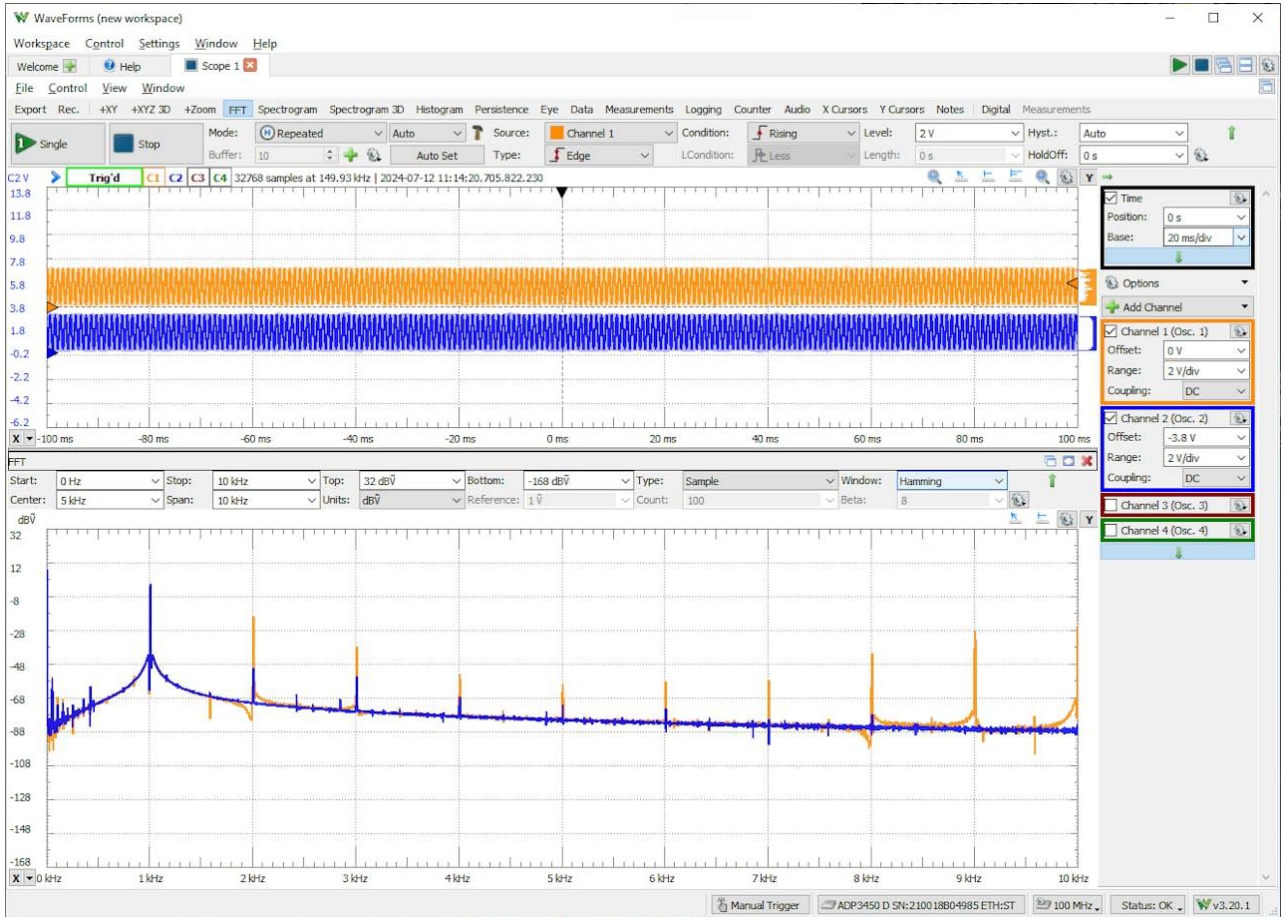
PWM Output

PWM Needed 2nd order RC Filter to reduce harmonics



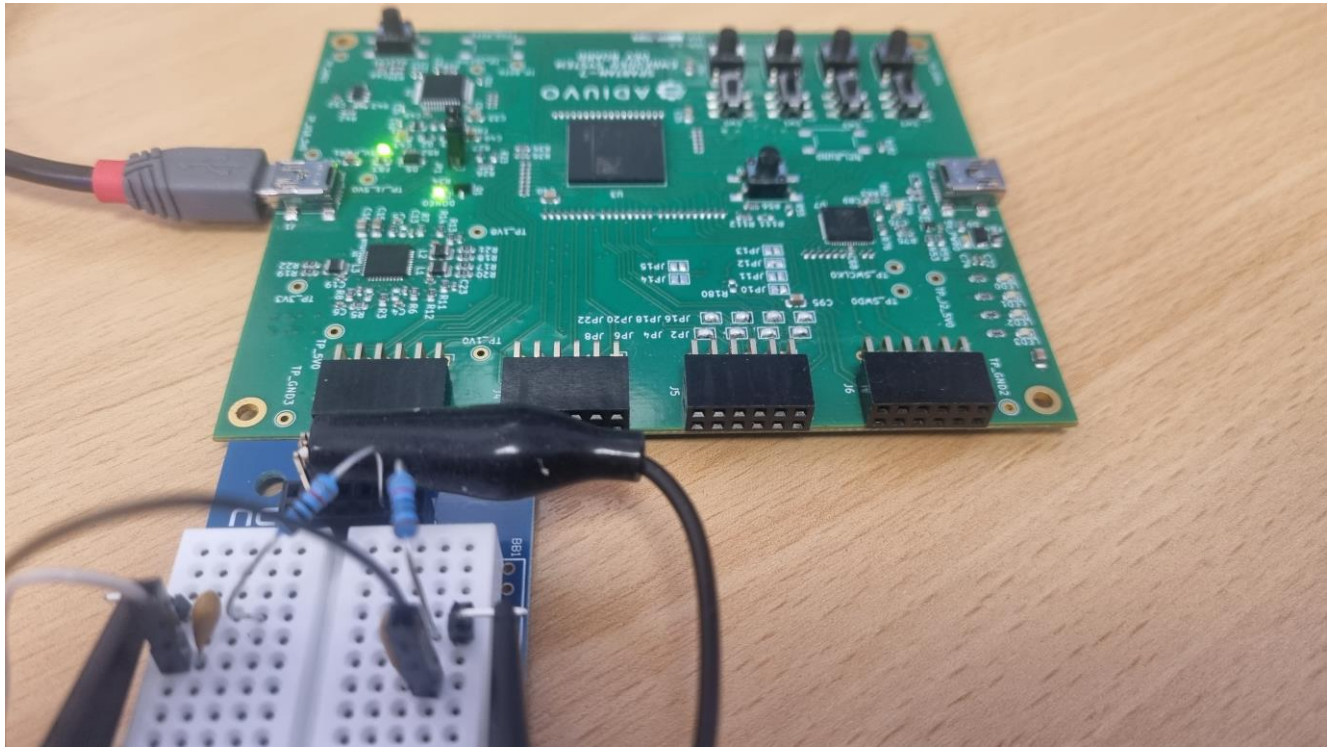
PWM Output

Comparison PWM vs Delta Sigma



Summary

- Working with Mixed Signal environment is not as difficult as might be expected.
- ADC and DAC characteristics and key parameters presented.
- We can even create simple systems using XADC / Sysmon and PWM / Delta Sigma DAC – using cost optimized portfolio devices.



ADIUVO
ENGINEERING AND TRAINING, LTD.

Questions

ADIUVO

ENGINEERING AND TRAINING, LTD.

www.adiuvoengineering.com

info@adiuvoengineering.com