**Product Test Plan (PTP)**

**SAR ADC Simulation, Processing, and Design**

**Author: Devin Santos**

**Date: 4/23/21**

1. Information Required
   1. Purpose
      1. The purpose of this project test plan is to describe in detail the tests that took place to verify the functionality of key components in the product. The overall goal is to prove by tests that the project functions as described. This document serves as a reference for anyone that uses this product.
   2. Scope
      1. The scope of this document includes the critical tests that were performed to validate each customer need. This document does not include the “hardware” tests that were performed since they do not relate to the customer needs. This project test plan includes the software related tests that are needed for the customer to use.
   3. Responsibilities
      1. As engineers, we are responsible for verifying the outcome of our products. Many applications need millions of iterations to test the functionality of the DUT. This document provides the minimum responsibility required by an engineer to validate this ADC processing product.
   4. References

|  |  |
| --- | --- |
| **No.** | **URL** |
| 1 | <https://training.ti.com/ti-precision-labs-adcs-coherent-sampling-and-filtering-improve-snr-and-thd> |
| 2 | <https://doc.xdevs.com/docs/_Books/ASIC_Design/cmos%20analog%20circuit%20design%20%28allen%2Cholberg-2002%29.pdf> |
| 3 | <https://www.maximintegrated.com/en/design/technical-documents/tutorials/1/1080.html> |
| 4 | <https://ieeexplore-ieee-org.mantis.csuchico.edu/document/8728470> |
|  |  |
| 5 | <https://www.researchgate.net/figure/SAR-Logic-Register_fig11_283721063> |
| 6 | <https://www.electronicshub.org/sample-and-hold-circuit/> |
| 7 | <https://pdfserv.maximintegrated.com/en/an/AN1080.pdf> |
| 8 | <https://github.com/PeterFeicht/ltspice2matlab> |
| 9 | <https://www.mathworks.com/help/matlab/ref/fft.html> |
| 10 | <https://microchipdeveloper.com/adc:adc-snr> |

* 1. Definitions
     1. Simulation: the production of a computer model of something, especially for the purpose of study.
     2. Product Test Plan: A test plan is a document detailing the objectives, resources, and processes for a specific test for a software or hardware product.
     3. Coherent Sampling: a method that increases the spectral resolution of an FFT and eliminates the need for window sampling when certain conditions are met.
  2. Equipment/Supplies
     1. Computer - <https://www.bestbuy.com/site/laptop-computers/all-laptops/pcmcat138500050001.c?id=pcmcat138500050001>
     2. LTspice - <https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html>
     3. MATLAB - <https://www.mathworks.com/downloads/>
     4. SeniorDesignProject.m - <https://github.com/devinpsantos/devinpsantos.github.io.git>
     5. LTspice2Matlab.m - <https://github.com/devinpsantos/devinpsantos.github.io.git>
  3. Precautions and Warnings
     1. CAUTION: Longer simulation times produce a large .RAW file that may require appropriate storage space in memory.
     2. WARNING: Follow the coherent sample equation precisely.

1. Testing Features and Functions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Customer Need** | **Functional Element** | **Physical Element** | **Software Element** | **Test Number** |
| Extract Desired Simulation Array | The generated .RAW file | 1D array of recorded data for desired LTspice node | LTspice  MATLAB | 1 |
| Extract Coherent Sampled Data | MATLAB code to keep desired samples using coherent sampling | For loops that pick which data samples are valid/desired from the array | MATLAB | 2 |
| Examine Frequency Spectrum | Perform FFT on the coherent sampled ADC output signal | Center and normalize to get correct FFT data array | MATLAB | 3 |
| Signal-to-Noise Ratio | Separate noise and signal samples from FFT data array | Square all values, sum all values, and take square root to plug into SNR equation | MATLAB | 4 |

* 1. Extract Desired Simulation Array (Test Number 1)
     1. Purpose: The user needs to be able to extract any desired variable and its data from the .RAW structure. Variable names are associated with the nodes in the spice model.
     2. Equipment Required: The required equipment for this test includes a computer, LTspice, and MATLAB.
     3. Input(s): The inputs for this test will be the .RAW file and the desired variable name entered by the user.
     4. Output Pass/Fail Criteria: For this test to pass the output must be the correct data variable and its contents that the user has requested.
     5. Test Description: The .RAW file will be entered into the LTspice2Matlab function provided. A list of all variables from the .RAW file will be displayed on the screen and the user will be prompted to enter the name of the desired variable to be examined. The code will then determine where the variable is and extract its data from the structure created by the given function. The data variable will be plotted for the user to compare the data to the simulation trace in LTspice. If the user examines that the plots match, then the test passes and if the data does not match the test fails.
     6. A picture containing long, line, different, chain

        Description automatically generatedTest Results:

Figure 2.1.5.1: This is the trace of “V(vout)” in LTspice after running the simulation and plotting the trace.

Text

Description automatically generatedFigure 2.1.5.2: This is the beginning of the MATLAB code created for this test. On line 9 the function LTspice2Matlab is being used to extract the data from the ideal\_adc.raw file and store it as a structure in the defined variable raw\_data. The rest of the code gets the time array, displays the variable names in file, and prompts to the user to enter the variable name in the command window.

Table

Description automatically generated

Figure 2.1.5.3: To the left is what gets displayed in the command window after running the code. The user picks what name to type and hits enter. The names that are displayed correlate directly to the LTspice simulation.

Text

Description automatically generated

Figure 2.1.5.4: After the user hits enter, in this case after typing “V(vout)”, it will display whether that variable has been found or not.

Chart, line chart

Description automatically generatedFigure 2.1.5.5: MATLAB will then display a figure for the user to check if the variable data matches the LTspice trace.

* + 1. Test Analysis: After comparing the two graphs with the plotted variable the data is a match and the test passes.
  1. Extract Coherent Sampled Data (Test Number 2)
     1. Purpose: The simulation data must be extracted using coherent sampling. To do this, specific data values are needed for certain step sizes in time. LTspice does not export its data with constant step sizes in time, so the variable data must be processed in MATLAB to achieve coherent sampling.
     2. Equipment Required: The required equipment for this test includes a computer, LTspice, and MATLAB.
     3. Input(s): The user will need to enter the number of windows, number of data samples, and their ADC sample frequency. The .RAW file will be used as another input to this test.
     4. Output Pass/Fail Criteria: The test will pass if the data elements to satisfy coherent sampling are extracted properly. The test fails if incorrect or no data elements are extracted.
     5. Test Description: Before running the code for this test the user must enter their computed value for the input frequency to the ADC under test. The user will also ensure the correct .RAW filename is entered in the LTspice2Matlab parameter. The user can then run the MATLAB script and examine the array that the data was put into. If there are missing values in any bin, then the test fails.
     6. Test Results:

Text, letter

Description automatically generated

Figure 2.2.5.1: This is the required equation and constraints for the user to choose a window and amount of data samples. The user will need to simulate their ADC in LTspice with the correctly computed input frequency.

Graphical user interface, text, application

Description automatically generatedFigure 2.2.5.2: In this section of code on lines 12 through 14 the user enters their values. The user also needs to enter the correct .RAW file name. After that, the MATLAB code gets the LTspice time vector and creates a new time vector that will achieve coherent sampling. Both arrays need to be rounded for the comparison between them to work. The LTspice data is rounded to the nearest 15 significant figures to the right of the decimal point and the desired time is 6 significant figures to the right. If this is not done the test will fail every time due to there being a very small difference in the numbers. I believe this is justifiable because it is still 0.001 picoseconds accurate.

Text

Description automatically generated

Figure 2.2.5.3: The way I find the correct data to match the correct time is to find the columns where the correct times are located. I create a nested for loop to check where the desired time matches the real time vector. When I find a match, I record that column number in a new array. This new array will then have all the correct Application, table

Description automatically generated with medium confidencecolumns to extract the data.

Figure 2.2.5.4: The user should then open their desired column vector by double clicking on it from the workspace window. If the loops matched the time correctly you will see a column number in every bin location. If there is a zero or NaN, then the test fails. To check for a passing test, the user should examine the time values at the recorded columns. If there is a match in time, then the program is working.

Text

Description automatically generatedFigure 2.2.5.5: After the desired columns are saved into an array the user enters the variable name as shown in Figures 2.1.5.3 and 2.1.5.4, another for loop extracts the correct columns from the raw LTspice variable data. On line 76 the desired columns array is used extract specific columns from “xt”, the user chosen variable. The array named “desired\_data” now only contains the coherent sampled data points.

Graphical user interface, application, table, Excel

Description automatically generatedFigure 2.2.5.6: After Figure 2.2.5.5 executes, the user should be able to open the desired data array and view its contents. There should be a value in every single bin. To check the correctness of the data extraction we can examine the plots.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Figure 2.2.5.7: On the left is a plot of the raw data that was extracted and on the right is a plot of the desired data that was extracted to satisfy coherent sampling. It can be examined that the time and amplitude data match up correctly, signaling that coherent sampling has been done successfully.

* + 1. Test Analysis: After checking the desired columns array and the desired data array, it is confirmed that the test has passed. If there was a missing bin value in either array the test would fail. Also, if there is a mismatch in time the test will fail. This process of getting column locations at specific time is required to attain only the coherent sample data points, as described by the equation specifications.
  1. Examine Frequency Spectrum (Test Number 3)
     1. Purpose: It is important that the correct FFT is captured to analyze the frequency spectrum. There should be a single spike at the input frequency value and an amplitude that is equivalent to the input signal’s amplitude.
     2. Equipment Required: The required equipment for this test includes a computer and MATLAB.
     3. Input(s): The input to this test will be a pure sinusoidal wave with a user specified frequency, amplitude, and DC offset.
     4. Output Pass/Fail Criteria: This test will pass if the spike has the correct frequency with the correct amplitude. It will fail if there is no spike, or the associated values are incorrect.
     5. Test Description: To test if the FFT calculations are correct, a perfect sinusoid will be generated by MATLAB. This sinusoid will then be evaluated in the frequency domain. The goal of this test is to verify that the FFT calculations are being performed correctly. By using a perfect sinusoid we are eliminating the chance of errors.
     6. Test Results:

Chart, line chart

Description automatically generatedText

Description automatically generatedFigure 2.3.5.1: This portion of code is creating a time array from the coherent sampling frequency specifications. The sine wave is then created using the defined time array, so the data samples have the correct step size.

Figure 2.3.5.2: This is the plot of the generated sine wave. It has an amplitude of 2.5 and a total of three cycles or windows.

Chart, line chart, scatter chart

Description automatically generatedText

Description automatically generatedFigure 2.3.5.3: This portion of code computes the FFT and FFT normalization. Extra steps are needed after line 27 to center the frequency and normalize the amplitude. The FFT signal is then plotted to visualize it.

Figure 2.3.5.4: This is the plot of the FFT after centering and normalizing it. There is a single spike at the correct frequency and amplitude of the signal.

Chart

Description automatically generatedFigure 2.3.5.5: This is the result when adding two sinusoids with a different frequency by a factor of three. We should now see two correctly placed spikes with an amplitude of 2.5 and 4 for the test to pass.

Chart, scatter chart

Description automatically generated

Figure 2.3.5.6: We can see that the FFT has two spikes associated with each individual sine wave. The test appears to have passed correctly.

* + 1. Test Analysis: After viewing the test results we can see the FFT calculations have been performed correctly. This test has passed, correctly displaying the frequency and amplitude of the signals for each test.
  1. Signal-to-Noise Ratio (Test Number 4)
     1. Purpose: An important ADC specification is its SNR. There is an equation that gives the most ideal SNR for an ADC. Most designed ADCs will fall short of the ideal value due to impurities in manufacturing and the design of the ADC sub-blocks. By computing the SNR of the user’s ADC, they are getting a specification feedback. Based on the value, the user can alter their design to achieve specification requirements.
     2. Equipment Required: The required equipment for this test includes a computer, LTspice, and MATLAB.
     3. Input(s): The inputs for this test will be the FFT array after it is correctly computed.
     4. Output Pass/Fail Criteria: For this test we will use an ideal ADC so we can compare the computed SNR to the ideal SNR value from equation 6.02\*N + 1.76. If the computed SNR is very close to this value, the test passes. If the computed SNR of the ideal ADC is not close to this ideal value, the test will fail.
     5. Test Description: To perform this test, an ideal ADC will need to be created in LTspice. Once that is done, the input frequency and simulation run time need to be computed for coherent sampling. After the simulation is complete the .RAW file needs to be copied and pasted into the MATLAB environment folder being used. The .RAW file will then be processed and the output from the ADC will be extracted following coherent sampling. The FFT will be computed to use the array for capturing the signal and noise values. These signal and noise arrays are then squared and summed to get the correct values for the SNR equation 20\*log10(signal/noise).
     6. Test Results:

Text

Description automatically generatedFigure 2.4.5.1: This is the first half of the LTspice netlist for the ideal ADC. I am setting it up as a 10-bit ADC to make it similar the SAR ADC I have already.

Text

Description automatically generatedFigure 2.4.5.2: This is the second half of the LTspice netlist to complete the ADC. There is an ideal 10-bit DAC to convert the digital value back to analog. This is done so MATLAB has voltage values to compute that are comparable to the input signal. Lines 85 and 87 show the labeling of the connections.

Chart, line chart

Description automatically generatedFigure 2.4.5.3: Here is the plot on MATLAB of the extracted output variable, vout, from the .RAW file after simulating the ideal ADC.

Graphical user interface, text, application

Description automatically generatedFigure 2.4.5.4: On line 142 I start a for loop that goes through every frequency bin to check which value matches the input frequency. I do this to find which column my signal amplitude is in. The next for loop finds which column the DC component is in.

Graphical user interface

Description automatically generated with medium confidenceText

Description automatically generatedFigure 2.4.5.5: On line 159 I start the loop one bin after the DC component and iterate through the FFT array until the end. If the column is not equal to my signal column, I add the data into the noise array. Once all the noise values are in the array, I square each bin and add them all together to get the total noise.

Figure 2.4.5.6: These are the results after computing lines 174 through 177. I justified the results by pointing out that the difference between my computed SNR of the ideal ADC is within %1 of the ideal equation value.

* + 1. Test Analysis: After performing the steps in test two and three, the data for the SNR computation becomes available. Using the FFT array and for loops, the signal and noise bins were separated. The final computed SNR value of the ideal ADC from LTspice came within %1 of the ideal value, 61.96. With this observation, the test has passed, the SNR computation code works. This should give the customer confidence that their ADC SNR is being computed correctly.

1. Other Testing
   1. User Interface
      1. Purpose: The user interface was created to allow the customer to enter their values after being prompted in the command line of MATLAB. The purpose of this interface is to give the user assurance that they are using the program correctly.
      2. Equipment Required: The required equipment is a computer and MATLAB.
      3. Input(s): The inputs for this test will come from the user. After reading the prompt they will type using their keyboard and hit the enter key to record their answer.
      4. Output Pass/Fail Criteria: After the user enters their information, the code will return a prompt telling the user if their answer is correct or incorrect. The deciding factor for answers being right or wrong for this test will be based on the coherent sampling specifications.
      5. Test Description: To test whether the interface works as expected I am going to enter some correct answers to see if they pass. I am also going to enter incorrect answers and make sure they fail. If an answer is not accepted the user will be asked to enter another answer until the answer is correct. The user must enter their ADC sampling rate %100 correct since there is no way to verify that answer.
      6. Test Results:

Text

Description automatically generatedFigure 3.1.5.1: This is the beginning of the .m file provided. On line 15 the user needs to enter the correct .RAW filename. Then the user can run the program and follow the prompts.

Graphical user interface, text, application

Description automatically generatedFigure 3.1.5.2: After hitting the run icon in MATLAB this is the first prompt the user gets. The user needs to enter their ADC sampling frequency that it was simulated with in LTspice. This value must match the spice ADC sampling frequency or the computations will be incorrect.

Graphical user interface, text, application

Description automatically generatedFigure 3.1.5.3: After hitting enter, the user will read what value has been recorded. The user will then be prompted to enter the odd number of windows chosen for coherent sampling.

Graphical user interface, text, application, email

Description automatically generatedFigure 3.1.5.4: If the user does not enter an odd number, they will keep getting asked to enter an odd number. The user will not be able to move on until an odd number has been entered. Keep in mind this number must be the same as the number they chose for the coherent sampling equation.

Graphical user interface, text, application, email

Description automatically generatedFigure 3.1.5.5: After entering the correct window size the code prompts the user into entering their chosen number of data samples from the coherent sampling equation. The program checks if the number is a power of 2 and returns to the user if the value is accepted. All numbers that are a power of 2 have only a single 1 in binary. If there is more than a single 1, then that number is not a power of 2.

Graphical user interface, text, application, email

Description automatically generatedFigure 3.1.5.6: The code will then calculate the required input frequency for the ADC and the spice simulation time. If the input frequency and run time are what the user entered in LTspice then they will type yes. If there is a mismatch in one or both numbers, then the user will type no. For this example, the numbers are incorrect, but I type yes anyways. This will show that if the user does not follow instructions the FFT and SNR computations will be incorrect.

Graphical user interface, text, application

Description automatically generatedFigure 3.1.5.7: After typing yes, confirming the input frequency and run time are a match to the LTspice simulation, the user will be instructed to enter the variable name they want to examine. In this case, my ADC output node in the spice simulation was called V(vfinal), so that is what I entered. If the variable is found they will be notified, and the code will start computing the results.

Graphical user interface, text, application, email

Description automatically generatedFigure 3.1.5.8: The user will then be prompted to enter the number of bits that their ADC is. My ADC is 10-bits, so I typed 10. Since in Figure 3.1.5.6 I typed yes when the values were incorrect, I get an invalid SNR value. Therefore it is important to follow instructions carefully.

Graphical user interface, text

Description automatically generatedFigure 3.1.5.9: If the user would have entered no, meaning the input frequency and run time are not matching their LTspice simulation values, they will be instructed to re-simulate with the correct values. They will then copy and paste the new .RAW file into the MATLAB working folder and run the program again. This time they will enter yes, and the correct results will be computed.

Graphical user interface, application

Description automatically generatedFigure 3.1.5.10: When being asked to enter the ADC output node name the user can enter an incorrect name. If the name is entered incorrectly the code will report that it could not find that variable and end the program. The user will then have to run the MATLAB script again and enter a valid name from the left column.

Graphical user interface

Description automatically generated with low confidenceFigure 3.1.5.11: This is the code at the end of the MATLAB script. If the variable name does not match or the LTspice simulation values are invalid, these prompts will output to the command window and end the program.

* + 1. Test Analysis: After going through the figures, it is obvious that the user can continue with the program with incorrect values. However, if the user is not malicious and follows the instructions, they will have a working program, computing correct results. This user interface cannot guarantee what the user enters is correct, but with certain verifications we can give the user assurance they are entering the right values for coherent sampling. Overall, I believe this user interface serves as a basic check that the entered values are correct.
  1. Installation Requirements
     1. LTspice: This is a free simulation software that was used for this project. It can be downloaded from the link <https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html>
     2. MATLAB: This is the software program used to take in the .RAW file and serve as the user interface. The FFT and FFTSHIFT commands are provided by MATLAB that allows for a fast computation. It can be downloaded from here <https://www.mathworks.com/downloads/>
     3. SeniorDesignProject.m: This is the MATLAB script I have created for ADC purposes. It performs coherent sampling, FFT, and SNR computations for the user. It can be found here <https://github.com/devinpsantos/devinpsantos.github.io.git>
     4. LTspice2Matlab: This is a MATLAB function that was created by Peter Feichtinger that is used to gather the .RAW file contents into an accessible structure. This file can be found here <https://github.com/devinpsantos/devinpsantos.github.io.git>
  2. Storage Requirements
     1. MATLAB requires anywhere from 4 to 6 gigabytes of storage to download. LTspice only takes up around 80 megabytes of storage to download. Additional storage will be needed for the MATLAB files and generated .RAW files. A computer with at least 10 gigabytes of free storage should be sufficient. To be extra cautious one may need around 20 gigabytes of free storage.