Implementation of Sigma-Delta ADC on Atlas DE0 Nano-SoC FPGA Board

Chinmay Gupta
Indian Institute of Technology Khargpur

Ashish Sharma
Inter-University Accelerator Center New Delhi

Abstract **- Sigma-Delta Analog to Digital Converters have a wide range of applications in the signal-processing domain for designing effective communication systems on hardware devices. This paper attempts the implementation of the converter by splitting the circuit into analog and digital part. The Input and Analog part are simulated through MATLAB and the Digital part is passed through Atlas DE0-Nano-SoC FPGA board using FIL (FPGA in the Loop Simulation). The initial input (delayed to match the output), the output digital approximation and the error (difference of the two signals) is observed through scope on MATLAB.**

Index Terms - Analog-to-Digital converters (ADC), Field programmable gate arrays (FPGA), FPGA-in-the Loop Simulation (FIL), Sigma Delta (∑Δ), System on Chip (SoC)

INTRODUCTION

The Sigma-Delta (Σ-Δ) ADC converter is a type of analog-to-digital converter that uses oversampling and noise-shaping techniques to achieve high-resolution and high-accuracy conversion. They have applications in multiple fields including audio signal processing, wireless communication systems and instrumentation and measurement instruments.

In layman terminology, an analog signal is a continuous-time signal i.e. information is encoded as a continuous variable. Whereas a digital signal represents data as a sequence of discrete values; at a given time it can take on one of a finite number of values.

The Sigma-Delta ADC works by oversampling the input analog signal at a significantly higher rate than the desired output digital signal. The oversampled signal is then passed through a 1-bit quantizer, and the resulting quantized signal is passed through a decimator to generate a high-resolution digital representation of the input signal.

The Sigma-Delta (Σ-Δ) ADC converter can be broken down into two major components, the Σ-Δ modulator and the digital/decimation filter [1].

The Σ-Δ modulator takes the analog input and converts it to a very high-rate bit stream which is then passed to the digital filter and decimator to convert the sampled data from the analog input to its digitized approximation. The modulator preferably samples the signal at a frequency much higher than the input frequency. The sampling frequency fs should be atleast two times the input frequency fi, according to Nyquist criteria to control aliasing effect of sampled signal. Aliasing is a phenomenon that occurs when a signal is samples at a very low rate, making it difficult to capture the original signal from it due to very high distortions.

Aliasing can be further reduced by taking the sampling frequency much higher than Nyquist frequency(fs=2fi). Apart from aliasing, running the modulator stage at a very high frequency has another added advantage of noise shaping. The quantization noise generated in the digitization of analog signal is generated at the high frequency of the modulator. This helps shift the low frequency noise to higher frequencies, outside the band of interest. This noise is later filtered using a low pass filter. Noise shaping is one of the reasons that DS converters are well-suited for low-frequency, high-accuracy measurements [1].

To summarize, high rate Σ-Δ ADCs are preferred over Nyquist rate ADCs for the several advantages they cater:

1.High Resolution: This high resolution allows for accurate representation of the input analog signal and improves the overall quality of the digital output.

2.Noise Shaping: It helps in reducing the quantization noise by oversampling the input signal and filtering it through a low-pass filter, increasing the SNR ratio.

3.Low Distortion: The high oversampling ratio and the use of high-order noise shaping filters minimize harmonic and nonlinear distortions, resulting in accurate reproduction of the input signal.

4.Simplified Anti-Aliasing: The oversampling frequency is much higher than the Nyquist frequency, reducing the requirement for anti-aliasing filters.

5.High Compatibility: This compatibility allows for advanced algorithms and digital filtering to be applied directly to the digitized signal.

SIGMA-DELTA MODULATOR

The Sigma-Delta modulator is presented with a time-varying input analog signal. The system clock implements the sampling speed fs, in conjunction with the modulator's 1-bit comparator [1] to produce high rate digital bitstream. As shown in Figure 1, the input voltage and the output of the 1-bit DAC are differentiated (Delta stage) and their output is passed through the Integrator. The Integrator accumulates this signal (Sigma stage) with the sum of the previous outputs of the Delta

stage. The updated sum (signal) is passed through a comparator (1-bit ADC), giving the bitwise error. This is channelled back to the delta stage through the 1-bit DAC, and is also passed on to the Digital Filter and Decimator stage.
The noise in the modulator is moved out to higher frequencies.
Figure 2 shows that the quantization noise for a first-order modulator starts low at zero hertz, rises rapidly, and then levels off at a maximum value at the modulator's sampling frequency (fs) [1].

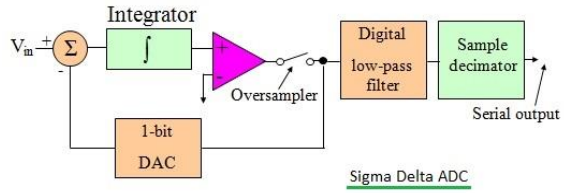

Figure 1: Block Diagram of Sigma-Delta ADC

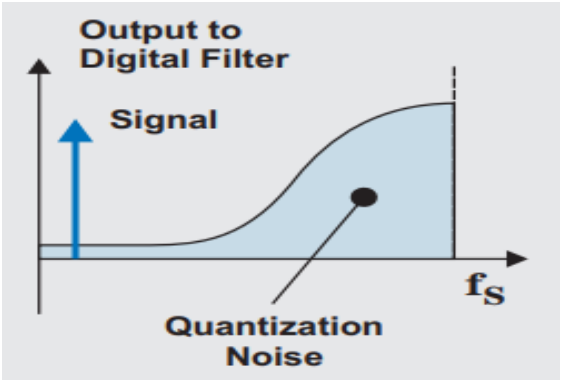

Figure 2: Noise Shaping by Σ-Δ modulator

DIGITAL FILTER AND DECIMATOR

The digital-filter function works as a low-pass filter by first sampling the modulator stream of the 1-bit code and removing the high frequency noise [2].
The output rate of the digital filter is still same as the modulator frequency fs. The output, though has a low quantization noise and high-resolution, is too fast to be processed properly by a controller or processor.
This signal can be brought down from frequency fs back to input frequency fi by using a process called down-sampling or decimation. Decimation is the process of reducing the sampling frequency of a signal to a lower sampling frequency that differs from the original frequency by an integer value. The frequency of the digital filter output is reduced by a factor of **(fs/fi)**, which results in loss of many intermediate samples, however no information is lost on passing a signal through a Decimator.
Putting both the Sigma-Delta modulator and Digital filter and decimator gives the overall Sigma-Delta ADC.

IMPLEMENTATION

The Sigma-Delta ADC comprises of Analog and Digital part. The Analog part contains the modulator which oversamples the input signal at a high rate and thus produces the output in the form of a binary sequence. Digital part of the converter consists of the FIR Decimation Filter, which down-samples the output of the Analog part to give a high resolution digital output signal. The decimation of the signal can be done in a single stage and can also be done in multiple stages.

A. MATLAB IMPLEMENTATION

The given circuit is figure 4 is implemented in MATLAB with an input sine wave and the results are plotted as follows:

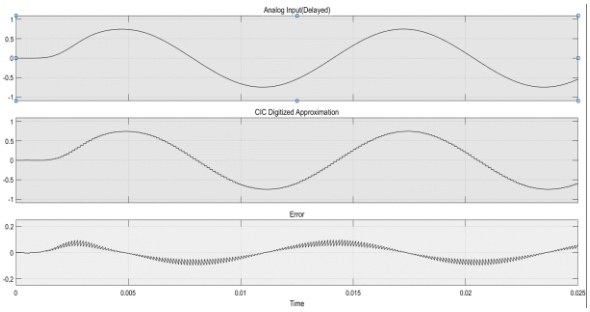

Figure 3: Input, Output and Error on Scope

The analog signal is first prefiltered using a low-pass Butterworth filter and the input signal is oversampled by a factor of 64 (fs=64*fi). The analog part consists of an Integrator, 1-bit quantizer and Zero-order hold blocks to form the Two-level ADC.
The feedback loop helps push the quantization noise generated by the ADC to higher frequencies (64*fs/2), away from the relevant signal band (fs being the input signal frequency) [5].
The CIC Decimation block performs a sample rate decrease (decimation) on an input signal by an integer factor. Cascaded Integrator-Comb (CIC) filters are a class of linear phase FIR filters comprised of a comb part and an integrator part [11]. The decimation stage down-samples the sampling rate back to the original input frequency of the signal. During this process, the quantization noise generated by the feedback loop are high-pass filtered beyond a frequency of fs/2.
Finally, the scope block shows the analog input signal, its digitized approximation and the error between the two signals.

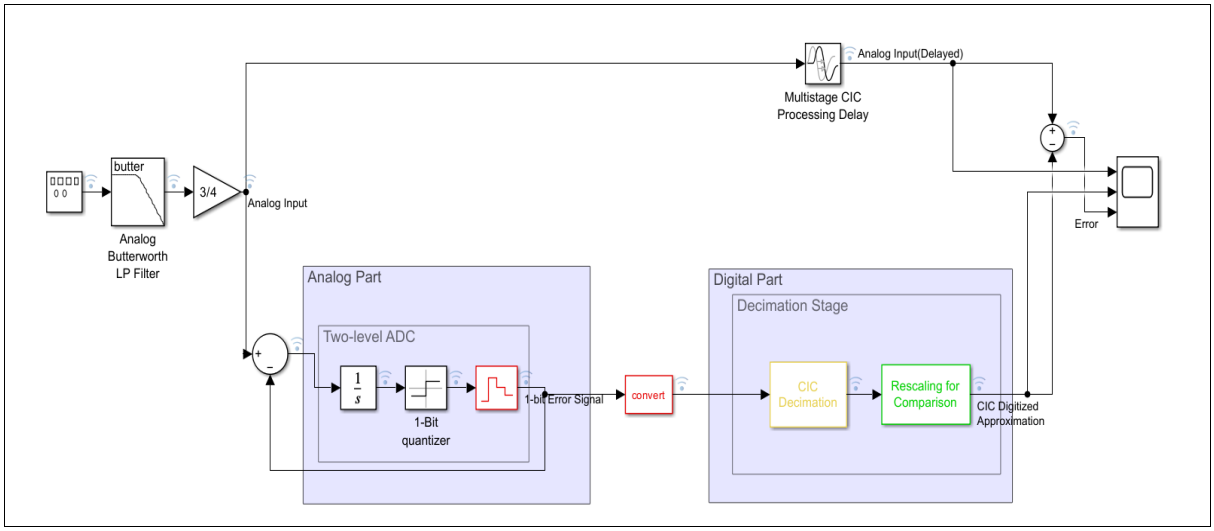

Figure 4: Simulink Implementation of Sigma-Delta ADC

B. FIL IMPLEMENTATION

For simulation of any component on Field Programmable Gate Array, it needs to be implemented in fixed point representation as floating point representation, though offers higher precision, it bears computationally expensive for hardware implementation. Therefore, we will implement the Digital part of the Σ-Δ ADC Converter in fixed point representation on Altera DE0 Nano SoC FPGA kit, co-simulated with SIMULINK using FIL, i.e FPGA in the loop simulation. SoC, also known as System on Chip refers to an integrated circuit or an IC that takes a single platform and integrates an entire electronic or computer system onto it. It is, exactly as its name suggests, an entire system on a single chip. The SoC on FPGA used in this implementation is of *Cyclone V family* and model *5CSEMA4U23C6N*.

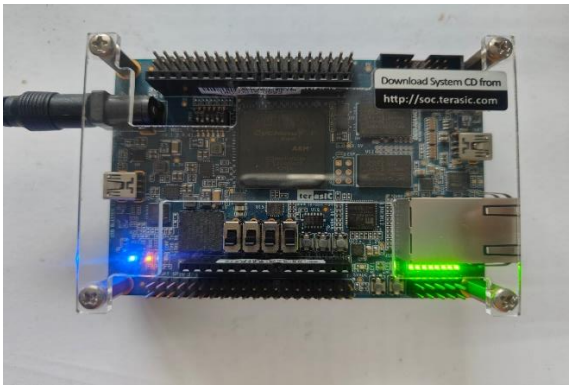

Figure 5: Altera Nano-SoC FPGA board

Steps for implementation of the converter using FIL are as follows:

1) Download and install Intel® Quartus® Prime software, version of 20.0 or lower as it is compatible with HDL Workflow Advisor.

2) Run the following command in MATLAB for path setup.

```
hdlsetuptoolpath('ToolName','Altera
Quartus II','ToolPath',...
'C:\intel\17.0\quartus\bin\quartus.e
xe');
```

3) Select the digital part i.e. the CIC Decimator and press CTRL+G to make it a subsystem. This subsystem is the target component for implementation in the FPGA.

4) Right click on the subsystem. Go to HDL Coder-> HDL Workflow Advisor. A window pops up.

5) Connect the FPGA board to the device and setup the Synthesis tool and Device as below and browse the Project Folder select Run this Task.

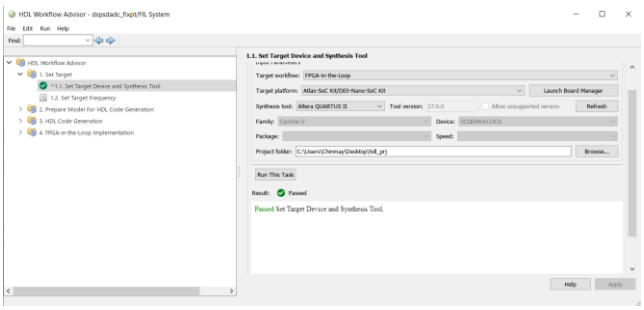

Figure 6: Set Target Device and Synthesis Tool

6) Set the Target Frequency to 50 MHz and Run the task.

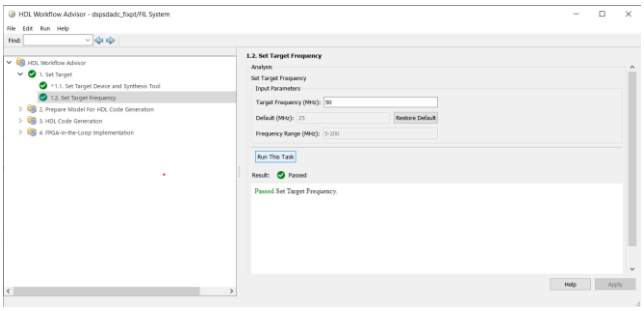

Figure 7: Set Target Frequency

7) Similarly, press Run all in HDL Code Generation tab and in case of warnings, select Modify all.

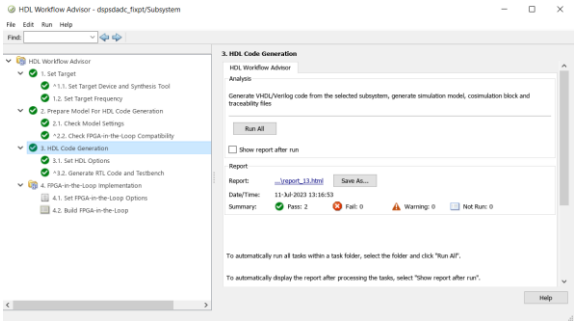

Figure 8: HDL Code Generation

8) Finally, run FPGA in the Loop tests and a Command window will prompt up.

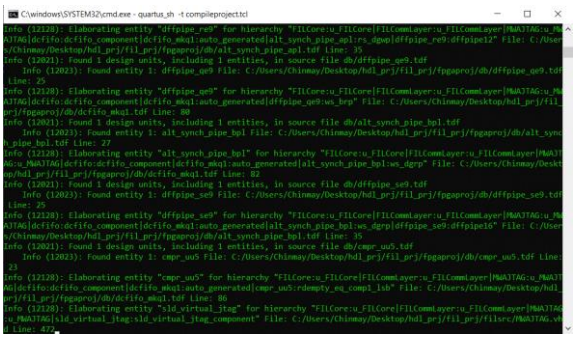

Figure 9: Command Window Prompt

9) A few seconds afterwards, a .slx file prompts in the MATLAB workspace.

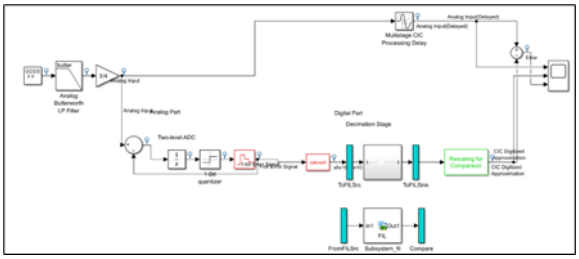

Figure 10: FIL

10) Click on 'Subsytem_FIL' and the Block parameters window opens up.

11) Browse the output .sof file from the Target location setup in Step 5 and press Load and then OK.

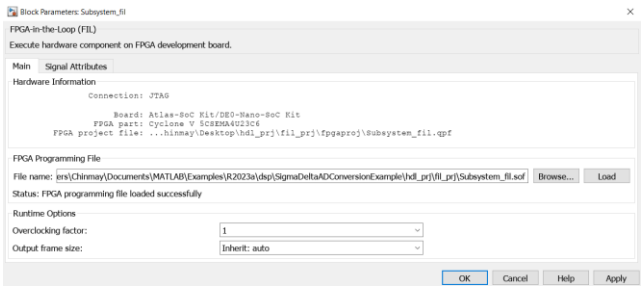

Figure 11: Block Parameters Window to load sof file

12) Finally Run the Simulation. The subsystem is computed in the FPGA board and the difference in FIL and MATLAB simulation results are compared in the compare scope.

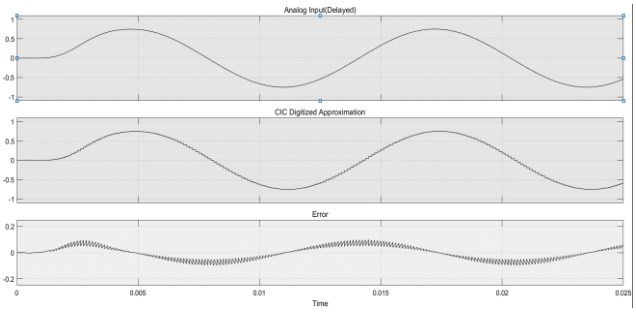

Figure 12: Input, Output and error of FIL simulation

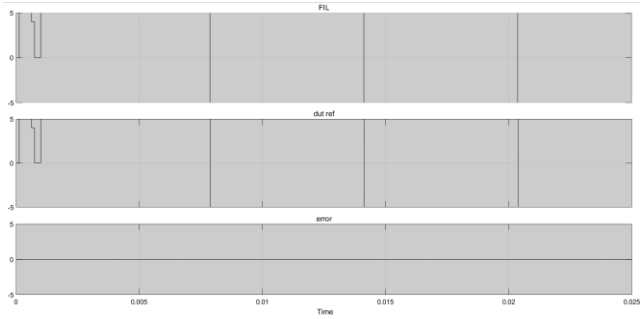

Figure 13: Comparison of DUT and FIL simulation results

CONCLUSION

In this paper, we have presented the design of a Sigma-Delta ADC converter and its implementation on DE0 FPGA kit. The fixed-point CIC decimator uses 5 sections is a good choice for hardware implementation and introduces a latency of 157.5 ($Latency = N \times \frac{(RM-1)}{2}$ samples, where N denotes the number of sections, M is the Differential Delay and R is the Decimation Factor) samples, rounded off to 158 samples [5], which is used as the time delay in the 'Multistage CIC Processing Delay Block'. This time delay helps to match the analog signal with its output digitized approximation for calculating error due to conversion.

Results show that the proposed design is quite accurate.

REFERENCES

[1] Bonnie Baker, “How delta-sigma ADCs work, Part 1”, Texas Instruments

[2] Bonnie Baker, “How delta-sigma ADCs work, Part 2”, Texas Instruments

[3] Simranpreet Kaur and Dr. Charanjit Singh, “FPGA Implementation of Sigma Delta ADC on Spartan 2”, International Journal of Advanced Research in Computer and Communication Engineering

[4] Daniel Piši, “FPGA based Delta-Sigma Analog to Digital Converter”, supervised by Ilona Kalová.

[5] https://in.mathworks.com/help/dsp/ug/sigma-delta-a-d-conversion.html

[6]https://in.mathworks.com/company/newsletters/articles/designing-a-sigma-delta-adc-from-behavioral-model-to-verilog-and-vhdl.html

[7] https://www.arrow.com/en/research-and-events/articles/engineering-resource-basics-of-analog-to-digital-converters

[8] https://www.analog.com/en/design-center/interactive-design-tools/sigma-delta-adc-tutorial.html

[9]https://in.mathworks.com/help/dsp/index.html?s_tid=CRUX_lftnav

[10] https://in.mathworks.com/help/dsp/fixed-point-filters.html?s_tid=CRUX_lftnav

[11] https://in.mathworks.com/help/dsp/ref/cicdecimation.html