Efficient Filtering Techniques based on Non-Uniform Signal Sampling and Processing

Capstone Project Proposal

Done by: Haneen Mohammed , Israa Alqassas , Reemaz Hetaimish and Sarah Alharthey

Abstract

This work aims to enhance the signal acquisition and processing chain required in remote systems. The motivation is to reduce their size, cost, processing noise, electromagnetic emission and especially power consumption. The classical or uniform sampling based techniques process the signal by employing a fixed order filter, which operates at a fixed sampling rate. Because of this nature, they are considered highly constrained in the case of low activity sporadic signals. To resolve this shortcoming, a filtering technique based on adaptive rate sampling is proposed. The focus is to achieve an efficient filtering technique by employing both uniform and non-uniform processing tools. This technique is based on EDS (Event-Driven Sampling), which adapts the sampling rate according to the input signal local variations and therefore reduces the processing activity. By employing the EDS features, an EDADC (Event-Driven Analog to Digital Converter) model has been developed to digitize a band-limited analog signal. The relevant active part of the non-uniformly sampled signal is selected and windowed by employing ASA (Activity Selection Algorithm). ASA displays interesting features with EDS, which are not available in the classical case. An offline reference FIR (Filter Impulse Response) filters bank is designed for the targeted application. The idea is to compare the results of the reference filtering with that of the proposed filtering. The proposed technique performance has been studied for a speech application. The goal is to determine the speakers gender. It can be done by measuring the speakers pitch. The computational complexity and the output quality of the proposed technique are compared to the classical one. Results show a drastic reduction in the total number of operations and therefore the power consumption.

Acknowledgement

Throughout the course of working on this research proposal for our Capstone Project we think it is due acknowledging the efforts of those who supported us. To Dr. Saeed Qaisar, our mentor and project supervisor, whom without, this project proposal would have struggled to be completed. To his constant support and guidance we say thank you. To the ECE department who would constantly inspire us to do our best, and encourage us to never complete a project until we are truly satisfied that is the best it can be. And of course to College of Engineering at Effat University who's many facilities and resources that were made available to us whenever needed, we thank you.

Contents

List of Figures

1 Introduction

1.1 Problem Statement

The mobile systems are becoming an essential part of our daily life. There are growing demands to enhance mobile systems in order to provide better services. Most of Efforts are concentrated towards improving the embedded systems design. However, very few works are focusing on achieving it with efficient signal acquisition and processing. Enhancing signal processing techniques often involves trade-offs because it is challenging to maintain the system size, cost, processing noise, electromagnetic emission and power consumption. The motivation of this project is to contribute to the development of smart mobile systems by achieving an efficient signal acquisition and processing techniques.

1.2 Literature Review

In digital signal processing systems, the ADC (analog to digital converter) is considered as a fundamental component. It affects the whole system performance [x]. Previous studies have compared different designs of ADCs to achieve high performance and power efficient signal processing systems [y]. One solution towards enhancing signal processing systems is the use of asynchronous ADCs, which do not require a clock [z]. An asynchronous ADC based on a binary tree structure has been studied and compared with other ADCs. Comparison results have shown that the developed 12-bit ADC achieves a very good trade-off between resolution, area, speed and power consumption [1]. Recent studies have shifted from the classical sampling techniques and have adopted Level Crossing Sampling Scheme, which is a non-uniform sampling technique that is based on the signal amplitude variations. An asynchronous analog-to-digital converter was designed by employing level-crossing flash technique. The designed ADC was compared with the previously designed ADCs. Results showed an important power consumption reduction [2]. An ADC based on Level crossing Sampling scheme has been designed for non-band limited signal reconstruction [3]. It was concluded that for some specific types of signals, level crossing sampling scheme has advantages over the classical one. Another work studied the spectral analysis of a signal driven sampling scheme by employing both uniform and non-uniform processing tools [4]. A spectrum analysis technique, which does not require oversampling, was presented. This technique utilizes the ASA (Activity Selection Algorithm) and the RFFT (Re-sampling + Fast Fourier Transform). Results have shown a significant improvement in terms of spectrum quality and computational complexity.

2 Proposed Solution

In classical scheme, for the case of low activity signals, a large number of samples are acquired and processed. It is a waste of acquisition and processing activity and therefore the power consumption. To resolve this shortcoming, a filtering technique based on adaptive rate sampling is devised. The focus is to achieve an efficient filtering technique by employing both uniform and non-uniform processing tools. This technique is based on EDS (Event-Driven Sampling), which adapts the sampling rate according to the input signal local variations and therefore reduces the processing activity by employing the EDS features, an EDADC (Event-Driven Analog to Digital Converter) model has been developed to digitize a band-limited analog signal. The active part of the non-uniformly sampled signal are selected and windowed by employing the ASA (Activity Selection Algorithm) [x]. ASA displays interesting features with EDS, which are not available in the classical case. The devised system block diagram is shown below.

2.1 Principle Block Diagram

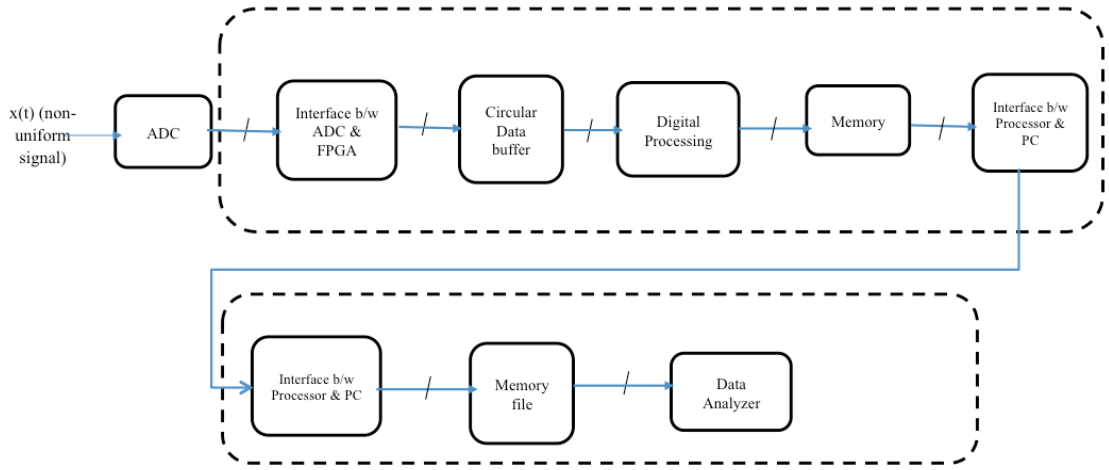

Figure 1: Block Diagram

The idea is to first study the system at algorithmic level. The behavioral model of the system will be designed and implemented with MATLAB. It will help in concepts illustrations and system simulation level verification. After a successful system MATLAB level implementation an interesting future work is to start developing the system critical modules architectures and implement them at material level. Because of its configurability FPGA is an interesting candidate in this context. It will be explained later in this proposal. Because of the commercially unavailable event-driven signal processing ADC, the sampling done is uniform using an oversampling technique with 200MHz when the needed frequency is only 8KHz. By doing so, the oversampled signal given to the processor would be a closer replica of the original analog signal. This is then resampled using the algorithm. Oversampling is the process of sampling the input signal at a significantly higher rate than Nyquist sampling rate, which is set at least as twice the input signal

frequency. In our system, the input signal x(t) which according to speech characteristics has the maximum frequency of 4KHz thus would require at least 8KHz to be properly sampled. With 200MHz sampling frequency instead of 8KHz, the over sampling rate (OSR) is:

$$OSR = \frac{Fs}{Fnyq} = \frac{200 \times 10^6}{8 \times 10^3} = \frac{200}{8} \times 10^3$$

The proposed filtering approach to be used in this project is The FIR Filtering Process. FIR filters are one primary type of digital filters used in Digital Signal Processing (DSP) applications. "FIR" means "Finite Impulse Response". If an impulse is given as input, that is, a single "1" sample followed by many "0" samples, zeroes will come out after the "1" sample has made its way through the delay line of the filter.

The time domain implementation of this filtering technique is given in the equation below: P is the filter order.

$$y_n = h_k \times x_n = \sum_{k=0}^{P} h_k.x_{n-k}$$

And for frequency domain implementation:

$$Y(f) = H(f).X(f)$$

Event driven filtering is used because only relevant number of operation to deliver per filtered output are taken into account.

2.1.1 ADC

Analog to digital convertors (ADC) are used in the conversion of analog signals into digital signals to allow Digital Signal Processing (DSP). In DSP systems ADC have a major impact on its performance therefore it is considered a critical component in the system. The conversion process starts by sampling the input analog signal (discretizing) and then rounding off these samples amplitudes (quantization) that will produce the quantized signal. Figure 2 shows a uniform deterministic quantization process that indicates that the reference thresholds are uniformly distributed.

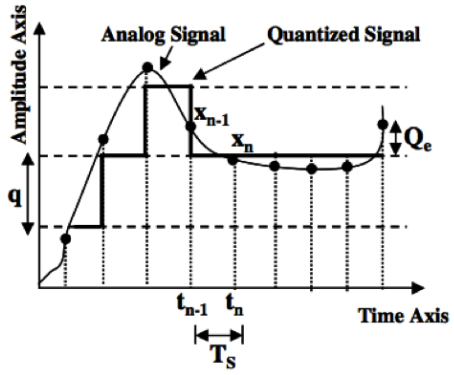

Figure 2: Uniform deterministic quantization process

A quantization error occurs in the classical ADCs because of analog input signals that may take any value within the ADC amplitude dynamics while the output is a sequence of finite precision samples. The number of bits defines the ADC resolution and Qe (t) defines the root main square of the error, which is,

$$RMS(Q(t)) = \frac{q}{\sqrt{12}}$$

Where q is the quantization step.

In uniform deterministic quantization the dynamic range of an M-bit resolution converter can be defined by

$$2^M = \frac{2 \times Vmax}{q}$$

ADC performance can be measured using mainly two parameters: SNR (Signal to Noise Ratio) and THD (Total Harmonics Distortion). SNR represent in decibel the ratio of RMS (root main square) desired signal amplitude to the RMS of noise amplitude as the following:

$$SNR(dB) = 20. \log \frac{RMS(Signal)}{RMS(Noise)}$$

The more the SNR value the better, as its an indication of the readability and reliability of the signal. Error in ideal ADC is the Quantization error and in practical ADC are clock jitter, the comparator ambiguity and other errors that occur in the conversion process. SNR can be measured theoretically and practically. Theoretically using the following equation

$$SNR(dB) = 6.92M + 1.76$$

And practically by applying monotone FS (full scale) sinusoid at the ADC input and by using the sequence of samples obtained at the converter output. Errors cause to reduce the resolution that is caused by the reduction of SNR. The resolution (ENOB: Effective Number of Bits) is computed with SNR of practical ADC

$$ENOB = \frac{SNR_{real}(dB) - 1.76}{6.02}$$

THD is ratio of the summation of harmonics power to the fundamental component power. Normally, the concerned harmonics are those lay in BWin. The process can be formally represented as follow.

$$THD = \frac{\sum HarmonicsPower}{FundamentalComponentPower}$$

This equation measure distortion that occure in signals which due to the nonlinearity of ADC. There are large varieties of classical ADCs that rely on the concept of uniform sampling and deterministic quantization. Two interesting types of ADCs that are suitable for this application are flash ADC and SAR ADC.

One of the fastest types of ADCs is Flash ADC. Flash ADC is suitable for applications that require large bandwidth and high speed and for low to medium resolution. Flash ADC block diagram shown in Figure 3.

Flash ADC architecture consists of series of high speed comparators that compare the input signal to the reference voltage where for M-bit converter, 2M̂-1 comparators are used. Reference voltage is provided by the resistive divider with 2M̂ resistors. Then the comparator output is connected to a 2M̂ M encoder that produces a binary output. This architecture of the flash ADC is referred to as thermometer code encoding because the design resembles a murcary thermometer. The operation of the Flash ADC is as the following:

- Vref for each comparator is one LSB greater than the reference voltage for the comparator below it.
- Comparator compare between Vref and Vinput , when Vinput ¿ Vref comparator produce 1 otherwise gives 0.
- Output of comparator then goes to 2M̂ to M encoder producing a digital output.

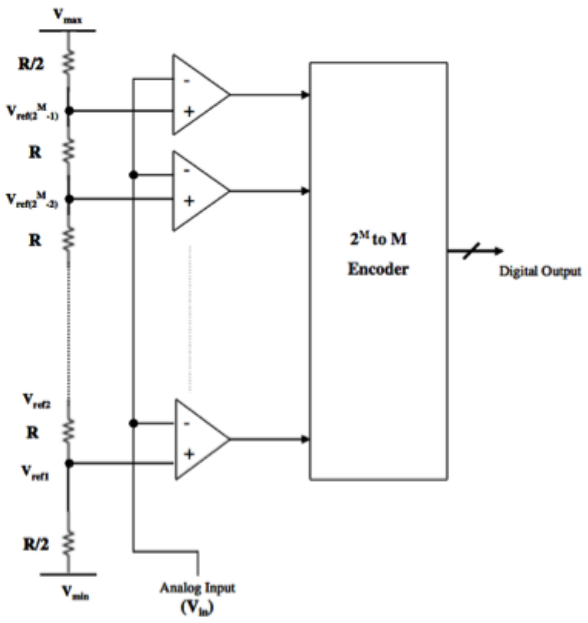

Figure 3: Flash

SAR (Successive Approximation Register) ADC is another type of analog to digital converter. It operates using a binary search algorithm and uses a successive approximation register that counts by going through all possible combinations starting from the most significant bit and finishing with the least significant bit. Its resolution ranges from 8 to 16 bits

The SAR ADC mainly consists of a comparator, DAC and a successive approximation register. An input voltage is fed into a comparator. The successive approximation register will start at the MSB, where it will be set to 1 (i.e 1000). The input voltage is compared with the output of the digital to analog converter.

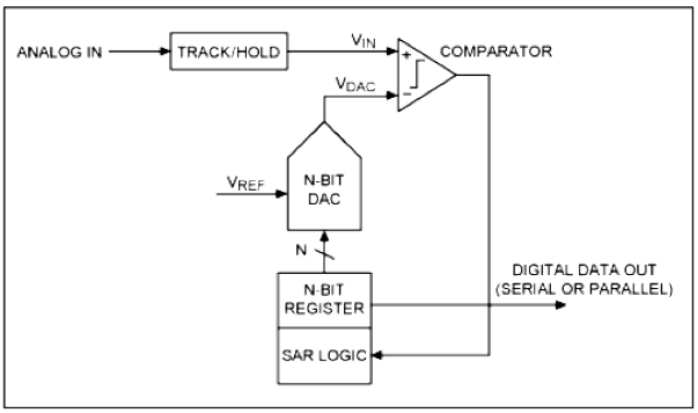

Figure 4: Successive Approximation Register

- If Vin is not greater than VADC, the output of the comparator will be 0 and will be fed to the successive approximation register. The MSB will be then reset to 0 and the pointer moves to the next bit and sets it to 1.
- If Vin is greater than VADC, the output of the comparator will be one and will be fed to the successive approximation register. The second bit will be set to 1 and the pointer moves to the next bit and sets it to 1.
- This comparison is performed all the way down to the LSB.
- Once the comparison is completed, the n-bit digital value will be stored in the register

Because a flash ADC uses a comparator for each bit, whereas the SAR ADC uses only one for the entire configuration, the design is considered simpler.

2.1.2 ADC to Processor interface

Interfacing ADC to a processor such as FPGA can be challenging. This is because ADCs use a variety of digital interface standards. Single data rate (SDR) is very common for lower speed data interfaces, typically under 200 MHz. In this case, data is transitioned on one edge of the clock by the transmitter and received by the receiver on the other clock edge (Applications Engineering Group, Analog Devices, Inc. 2013). This provides enough time for the data to dispose before sampling. In double data rate (DDR), Figure 5,

the transmitter transitions data on both clock edges (rising and falling). This allows for twice as much data to be transferred in the same amount of time as SDR (Applications Engineering Group, Analog Devices, Inc. 2013). Less time however is provided to the data to settle before sampling. It divides the number of physical channels by 2 hence allowing more data to pass in the same amount of time and this give it its advantage. There are some basic pins on an ADC that are used for interfacing. These are:

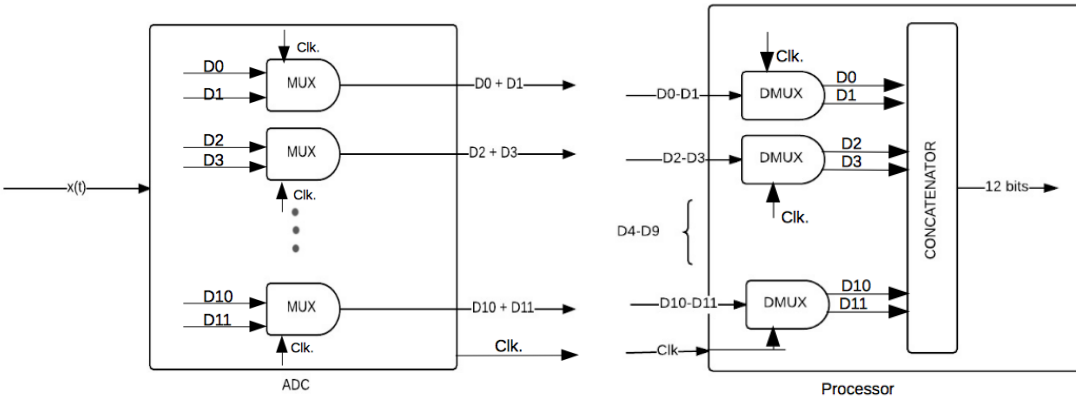

Figure 5: Double data rate (DDR)

- MODE: determines whether the data is being read (1) or written (0).
- LOAD: used after a write command to place the data into the internal registers.
- DATA: is a bidirectional signal as data flows in both directions.
- CLK: signal clocks the data into and out of the ADC

When connecting the ADC to a processor, as an FPGA, 12 bits carry 12 signals that can be transmitted to the FPGA using 2 types of wiring: Single ended Figure 6, and Differential, Figure 7.

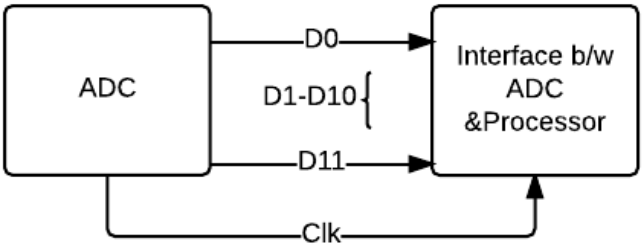

Figure 6: Single Ended

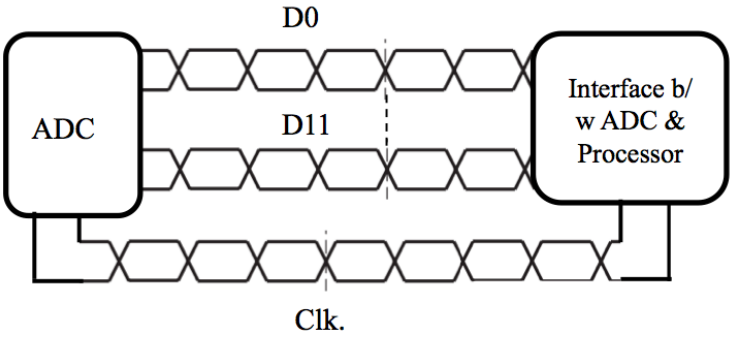

Figure 7: Differential input

In this system low voltage differential wiring is used because it is more immune to noise. 12 signals, 2 wires each gives a total of 24 wires. A differential voltage is floating, meaning that it has no reference to ground. The measurement is taken as the voltage difference between the two wires. The main benefit of a differential measurement is noise rejection, because the noise is added to both wires and can then be filtered out by the common mode rejection of the data acquisition system. Differential measurements should be used if the sensor is in a noisy environment or for sensors with output voltages susceptible to noise interference. In low voltage differential signals (LVDS), the bit clock rate for two-wire interface:

$$\frac{ADCResolution \times SampleRate}{2} = BitClock(MHz)$$

$$\frac{\frac{ADCResolution}{2} \times SampleRate}{2} = BitClock(MHz)$$

In this project:

$$\frac{\frac{12bits}{2} \times 200MHz}{2} = 600MHz)$$

The two-wire reference design (featured ADC interface) supports ADCs with sampling resolutions of 12, 14, or 16 bits, and sampling speeds of up to 125 MSPS for 16 bits. The modular design approach of both reference designs allows for easy modification to support higher resolutions and/or a different number of channels Defossez, M. (2008).

Each of the 2 types of wiring can have the clock be synched either using source or sink.

- In a sink synch approach the processor receiving the signal does not provide power and acts as a load that is connected to the ADC.
- In a source synch approach the processor receiving the signal provides the power to the ADC.

Figure 8 below shows how the connection between the processor and ADC is done in each of these 2 approaches:

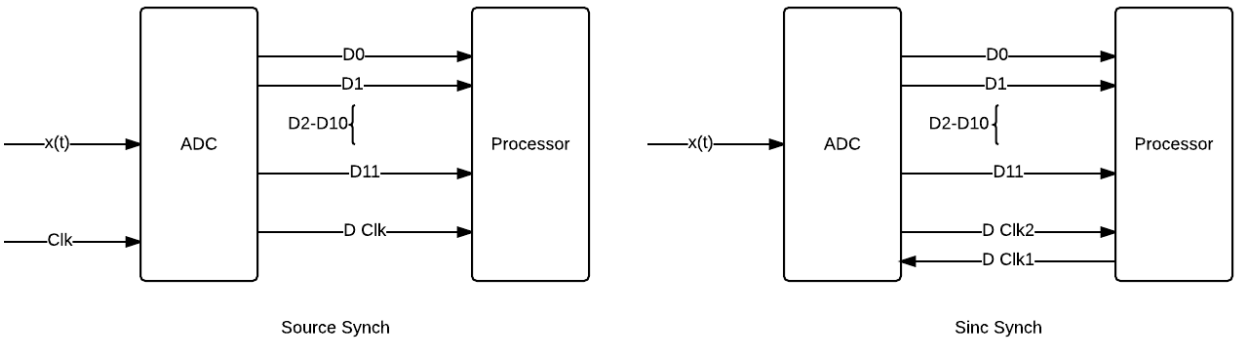

Figure 8: Source (left) Sinc (right)

2.1.3 Circular buffer (FIFO)

The sample outputs then become the input to the function that stores the sample signals in a data buffer that acts as a memory. The samples must be continually updated as new samples are acquired. For this purpose, a circular buffer is used. Circular buffers are commonly used in embedded systems to provide synchronization between the data and the clock. In other words, they are used when the application requires minimal delay for the information to be immediately available. Data in a circular buffer are stored as a first-in first-out sequence. When the buffer is full, the old data are overwritten with the new data. The advantage of using circular buffers is not having to use a memory of large capacity because data are automatically overwritten. Circular buffers are considered more efficient than linear buffers because only one value has to be changed when a new sample is acquired [5]. Hence, reading and writing operations can take place simultaneously. FIFO (First In First Out), which is a common implementation of the circular data buffer, is used to store the sampled signal temporarily before being processed.

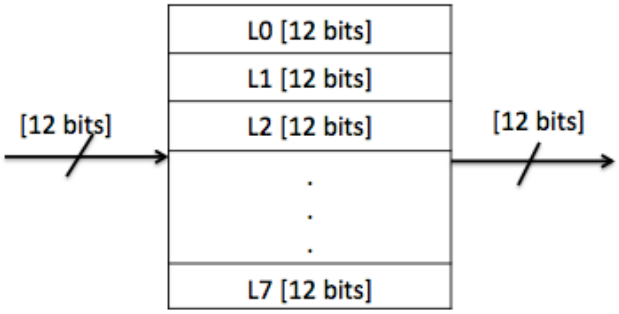

Figure 9: FIFO

2.1.4 Digital Signal Processing

The digital signal is processed using Level Crossing Sampling Scheme, which is also known as Event Driven Sampling. This technique adapts the sampling rate according to the input signal local variations. The samples are recorded only when the signal crosses one of the predefined reference levels. The advantage of using this technique is to process only the relevant information, which results in producing less samples and thus less processing activity. By employing Event Driven Sampling features, a model will be developed to digitize a band-limited analog signal. The signal will be band-limited using a low pass filter. Figure 10 illustrates the Event Driven Sampling on a circuit level. A Level Crossing Detector (LCD) will be developed to detect the points at which the signal crosses predefined levels. A Timer Circuit with a clock is developed to measure the timing instants and keep track of the changes in time between each instant. These timing instants are stored in a register.

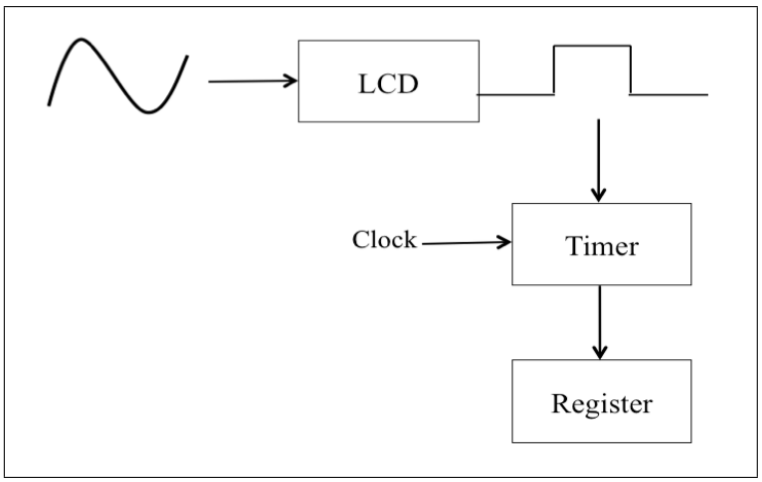

Figure 10: Event Driven Sampling Block Diagram

The relevant active part of the non-uniformly sampled signal is selected and windowed by employing the smart windowing process. The employed windowing technique displays interesting features with EDS, which are not available in the classical case.

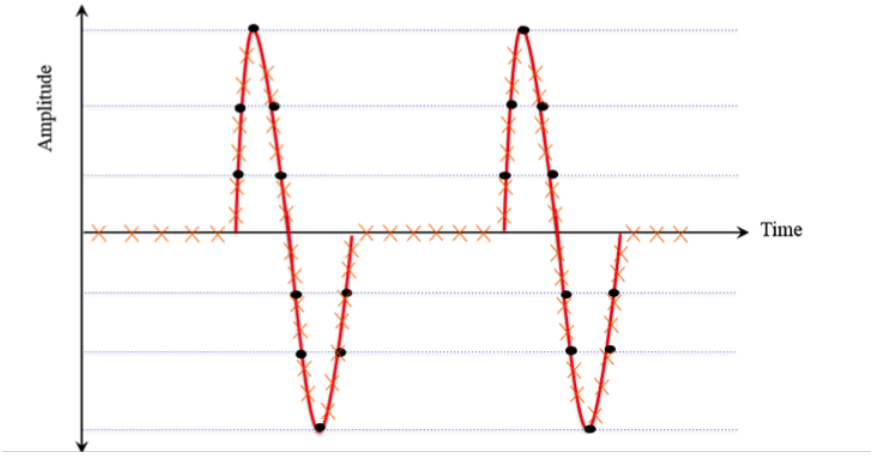

Figure 11: EDADC

2.1.5 Dual Port Ram

The values of the processed signal get then saved in a dual port RAM (Random Access Memory) ready to be transferred to a PC to be used in various applications. Dual port RAM is a type of memory that can read and write different memory cells simultaneously at different addresses. Figure 12shows the block diagram for the memory.

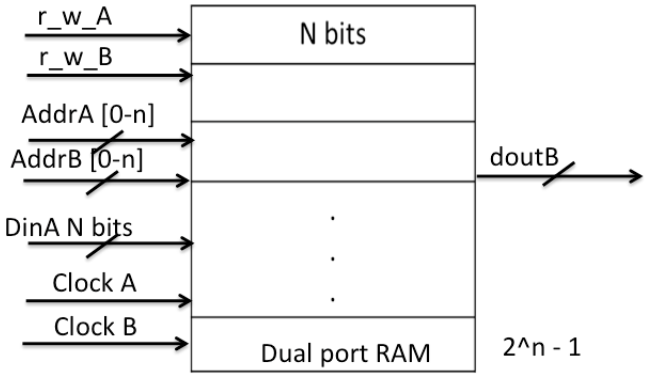

Figure 12: Dual Port Ram

2.1.6 Processor to PC interface

The sampled signal then would be transferred to a PC. To do that, processors need a way to communicate with the PC. There are many ways for data exchange between two devices and these include serial or parallel, and synchronous or asynchronous techniques, and simplex, full duplex, or half duplex. In parallel communication, the data is sent in blocks of bytes at the same time. On other hand, serial communication sends data one bit at a time. As for the synchronization, a system with shared clock between the receiver and the transmitter is a synchronous, and a system without a common clock is an asynchronous. Instead of sending a clock signal to the receiver, an asynchronous communication uses an agreed-upon timing parameter and special bits to synchronize the transmission process, Figure 13.

Start Bit	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Stop Bit	Next Start Bit	Next Data frame

Figure 13: 8 bit word length Frame

The mostly used component that translates data asynchronously between parallel and serial forms is called a UART, which stands for Universal Asynchronous Receiver/Transmitter. The UART takes bytes of data and transmits the individual bits sequentially at a specified data rate. At the destination, a second UART re-assembles the bits into complete bytes. This method of serial communication is referred to as TTL serial (transistor-transistor logic). Serial communication at a TTL level will always remain between the limits of 0V and Vcc. UART does not directly generate or receive the external signals used between different items of equipment. Separate interface devices are used to convert the logic level signals of the UART to and from the external signaling levels. There are different standards that define these external signals. The serial port on most hosts PCs complies with the RS-232 standard. The standard defines the electrical characteristics and timing of signals, the meaning of signals, and the physical size and pin-out of connectors. RS232 electrical specifications are documented in the EIA (Electronics Industry Association).

To use the USB port on the PC, an USB to UART Bridge is used. it is an Electronic device that provides USB connectivity to UART peripheral between the embedded system and the host PC. Figure 14. shows a block diagram of a system that incorporates the USB - to - UART bridge. The embedded system connects to the bridge through a standard UART interface. The PC connects to the bridge through the USB port. In a typical USB to UART bridge the host PC recognizes the bridge as virtual port. Finally, the sampled data sent will then be stored in a data file in the PC.

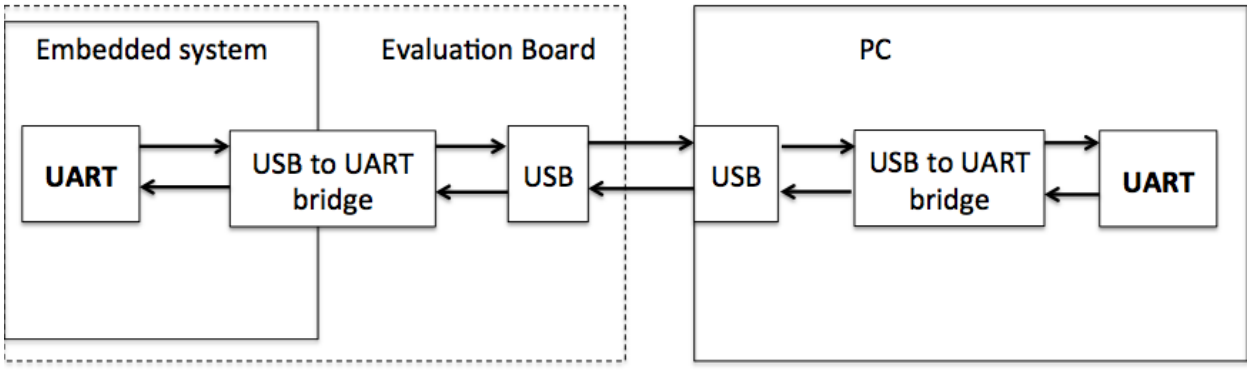

Figure 14: Processor to PC interface

2.2 System Design and Implementation

The design will be studied on a software-based level. MATLAB is a good candidate to analyze and test the implementation of the proposed design. Once the design is proven successful, it is planned to implement it using an embedded hardware system.

2.3 System Specifications

Part Number	Description	Specifications	Unit Price
AD9613-250EBZ	Evaluation Board for the selected ADC	**Number of Bits: 12** **Sampling Rate: 250M** **Data Interface: SPI** **# of A/D Converters: 2**	326.25$
CVT-ADC-FMX-INTPZB-ND	Interposer (Interface between ADC and embedded system)	Related Products: Xilinx FMC-Supported Boards	114.55$
EK-S6-SP601-G	Spartan-6 FPGA SP601 Evaluation Kit	200MHz Oscillator (Differential), Serial (UART) to USB Bridge, DDR2 Component Memory 128MB, IIC 8Kb IIC EEPROM, FMC-LPC connector (68 single-ended or 34 differential user defined signals)	338.75$

2.4 Proposed Implementation Plan

Task	Deadline
Develop the algorithm for event driven sampling	Mid- March
Develop the MATLAB code for event driven sampling	Mid- March
Test the MATLAB code	Mid- March
Purchase the hardware needed for implementation	Mid- February
Implement on hardware level	May
Finalize Project, Project Report and Presentation	End-May

3 Project Risk Management and cost optimization

As the project is software based, no cost will be needed to perform the project, yet if the project will be hardware implemented, the cost of the project is an essential point of consideration in the design phase for the application proposed. Most of the cost is by the purchased components for hardware implementation. It is found that one of the main challenges is the availability of the components as well in the Saudi market like FPGA is advance and couldnt be found in regular electrical supplies shop. Therefore, the components will be purchased and ordered online through multiple authorized websites taking into account the price variation of each component and the shipment time.

4 Conclusion

An event driven filtering technique is to be created to contribute to the development of remote systems and smart mobile systems by achieving an efficient signal acquisition and processing techniques. The implementation will be on speech recognition application taking into account the voice input signal characteristics and the application requirement. The proposed filtering technique which is based on non uniform sampling result to a reduction in size, cost, processing noise, electromagnetic emission and especially power consumption. This achieved with join benefits of EDS (Event-Driven Sampling), EDADC (Event-Driven Analog to Digital Converter) and ASA (Activity Selection Algorithm) by taking benefit of the input local variation. Therefore, in the speech recognition application, only innovative signals will be sampled and processed through the proposed circuit design and evaluated to showcase the delivered output quality and the significant gain in the computational load compered to classical techniques.

References

[1] Filipp Akopyan, Alyssa Apsel, and Rajit Manohar. Asynchronous analog-to-digital converter and method, December 16 2008. US Patent 7,466,258.

[2] Filipp Akopyan, Rajit Manohar, and Alyssa B Apsel. A level-crossing flash asynchronous analog-to-digital converter. In *Asynchronous Circuits and Systems, 2006. 12th IEEE International Symposium on*, pages 11–pp. IEEE, 2006.

[3] Karen M Guan and Andrew C Singer. A level-crossing sampling scheme for non-bandlimited signals. In *ICASSP (3)*, pages 381–383. Citeseer, 2006.

[4] Saeed Mian Qaisar, Laurent Fesquet, Marc Renaudin, et al. Spectral analysis of a signal driven sampling scheme. In *Proceedings of the 14th European Signal Processing Conference (EUSIPCO'06)*, 2006.

[5] Steven W Smith et al. The scientist and engineer's guide to digital signal processing. 1997.