Implementation of FIR Filter using Distributed Arithmetic Method

A thesis submitted in partial fulfilment of the

Requirements for the degree of

Bachelor of Technology

In

Electronics and Instrumentation Engineering

By

Smruti Sarita Swain

Roll no-111EI0382

Under the guidance of

Prof. Ayas Kanta Swain

Department of Electronics and Communications Engineering

National Institute of Technology, Rourkela

CERTIFICATE

This is to certify that the thesis entitled **"Implementation of FIR FILTER using Distributed Arithmetic method"** submitted by Smruti Sarita Swain (111EI0382) in partial fulfilment of the requirements for the award of BACHELOR OF TECHNOLOGY Degree in Electronics and Instrumentation Engineering at the National Institute of Technology, Rourkela is an authentic work carried out by her under my supervision and guidance.

To the best of my insight, the matter exemplified in the proposal has not been submitted to some other University/ Institute for the grant of any degree or certificate.

Date: 11th May, 2015

Prof. Ayas Kanta Swain
Assistant Professor
Department of Electronics and Communications Engineering
National Institute of Technology
Rourkela

ACKNOWLEDGEMENT

We benefit this chance to express our obligation to our guide Prof. Ayas Kanta Swain, Department of Electronics and Communications Engineering, National Institute of Technology, Rourkela, for his significant direction, consistent consolation and kind support at different stages for the execution of this paper work.

We additionally express our earnest appreciation to Prof. Kamalakanta Mahapatra, Head of the Department, Electronics and Communications Engineering for permitting access to significant lab facilities in the division.

Furthermore, last yet not the minimum, we are grateful to each one of those companions who have helped us over the span of this whole thesis work.

Smruti Sarita Swain(111EI0382)
Department of Electronics and Communications Engineering
National Institute of Technology
Rourkela

ABSTRACT

The prime objective of this project is to implement FIR filter using Distributed Arithmetic method. In order to perform the filter computation, it is a prime requirement to process real world signals and extract information from them. Hence ADC interfacing is essential since it converts analog world to digital domain so as to make it suitable for digital operations. The FIR Filter is simulated using Distributed Arithmetic method and is simulated in XILINX ISE Design Suite 14.2 and ADC is interfaced with Spartan 3e FPGA board.

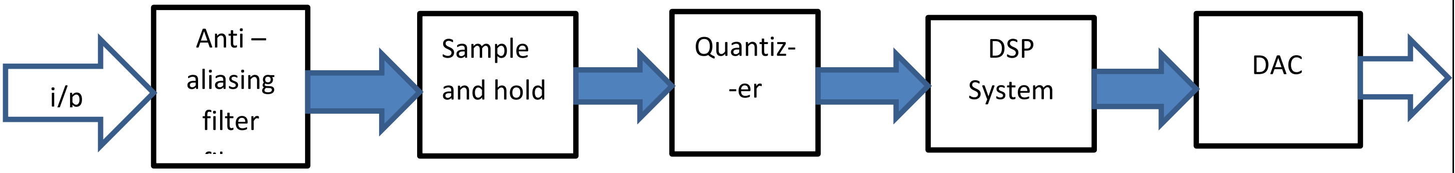
Afterwards, digital output is converted into its analog counterpart via DAC interfacing with the same Spartan 3e board.

CONTENTS

CHAPTER 1: INTRODUCTION

The prime task in Digital Signal Processing (DSP) is modifying a signal mathematically to improve it and make different operations easier for it. It measures, compresses or filters real world analog signals. In current world, Digital Signal Processing (DSP) has become a necessary technology and has taken the place of many traditional analog signal processing systems. There are several advantages of DSP systems such as indifference to fluctuations in temperature, aging effect and component tolerance. Another merit of digital systems is that digital designs can be more densely integrated than analog designs as inferred from past years.

Technological developments have caused an acceleration in the development of the area of DSP. One of them is the devising of an efficient algorithm to calculate the Discrete Fourier Transform. The introduction of programmable digital signal processors (PDSPs) in the late 1970s was another major step in this field. These were able to perform “Multiplication and Accumulation” (MAC) in one clock cycle . It was an important improvement in the “Von Neumann” microprocessor based systems in those years. More improved functions such as memory bank, floating point multipliers, or zero overhead interface to ADC and DAC are included in the modern PDSPs. Digital Signal Processing finds major use in decoding, coding, image compression , audio and speech processing .



1.1 OBJECTIVE

- The major objective of this paper is to reduce the computation in designing a FIR filter and hence the hardware resources required for that by devising a novel method called Distributed arithmetic method.

- To implement the above design on Field Programmable Gate Arrays (FPGAs), it is necessary to input the signal given by us and transform it into digital domain through ADC.
- Finally to use the transformed data in filter design and convert it back to analog format through DAC present on the board.

1.2 Software and Hardware utilised

The hardware used is Xilinx Spartan 3E FPGA board with present on the board ADC (LTC 1407) and DAC (LTC 2624). The software used is Xilinx ISE Design Suite 14.2.

1.3 Testing Technology Used

ChipScope Pro analyser is used for testing purpose since it makes easier to see the signals generated internally in the FPGA board. These signals cannot be seen with the help of an oscilloscope since the signals produced by the ADC do not have interface to outside. To visualize the signals of the DAC, a DSO (Digital Storage Oscilloscope) is used, and a function generator is used to give the input signal to the ADC.

1.4 Applications

The various applications of ADC and DAC are as listed below:

- Digital Signal Processing Applications. Analog signals provide users with immense information that is useful in one way or the other. Various functions are applied to those signals which help us to extract those information embedded in the signals.
- Music and data recording. Sampling the data in such cases is of prime importance for maximum performance and hence ADCs and DACs come into the picture.

The ADC used here is Successive Approximation Register (SAR) ADC i.e the on-board ADC of Spartan 3e kit. It helps in many fields of DSP, instrumentation and communication and many other ones.

CHAPTER 2:

ADC INTERFACING

2.1 **INTRODUCTION**

The salient features of the ADC chip are as follows:

- 1.5Msps (Mega Hertz Sampling rate) Throughput per Channel .It indicates the efficiency of the ADC chip and says how exactly and efficiently it can convert analog data into its digital counterpart.

- 3 Msps Sampling rate of ADC. The rate of sampling is divided into two channels each having sampling rate of 1.5 Msps. Here the term Msps denotes the conversion rate of ADC used.

- Low Power Dissipation: about 14mW. Very low power is dissipated and hence it can be ideally used for chip level applications. The hardware level implementation is also suitable as the power consumption is very less and given the circumstances demanding , a low power consuming mode can also be switched onto.

- Pin Compatibility 0V to 2.5V Input Range Version (LTC1407A). The pins are inherently compatible to an input range of 0V to 2.5V. Any range outside this is has to be processed before giving it to the chip.

- Input Range-.±1.25V. The DC ranges from is +1.25V to -1.25V. It can be manipulated in the way it has to be behaved. By changing the gain parameters input voltage range can be set by programmer.

- 3V Single Supply Operation. Inputs are taken from a supply of 3V DC.

- 2.5V Internal Band gap Reference with external overdrive. Internal Band gap i.e. the difference between its two states is around 2.5 V and this is driven externally from an outside source.

- 80dB Common Mode Rejection at 100 kHz. This parameter finds importance whenever simultaneous sampling of both channels of the ADC. Noise being generated is minimized by common mode rejection and the desired output is obtained.

- 3-Wire Serial Interface. The external world is interfaced serial bus protocol. Here, it is SPI- Serial Peripheral Interface bus.

Applications of this ADC chip :

- Telecommunications: Most of the operations is done in digital numbers in current scenario in this area. So it is necessary to convert analog data to digital for ease of operation.

- Data Acquisition: ADC is used in SCADA (Supervisory Control and Data Acquisition) that uses coded signals for communication. It requires an efficient ADC for efficient acquisition and retrieval of data.

- Uninterrupted Power Supplies (UPS): Digital format is suitable for charging purposes rather than analog . An UPS requires an ADC for proper battery usage and efficient storage of power.

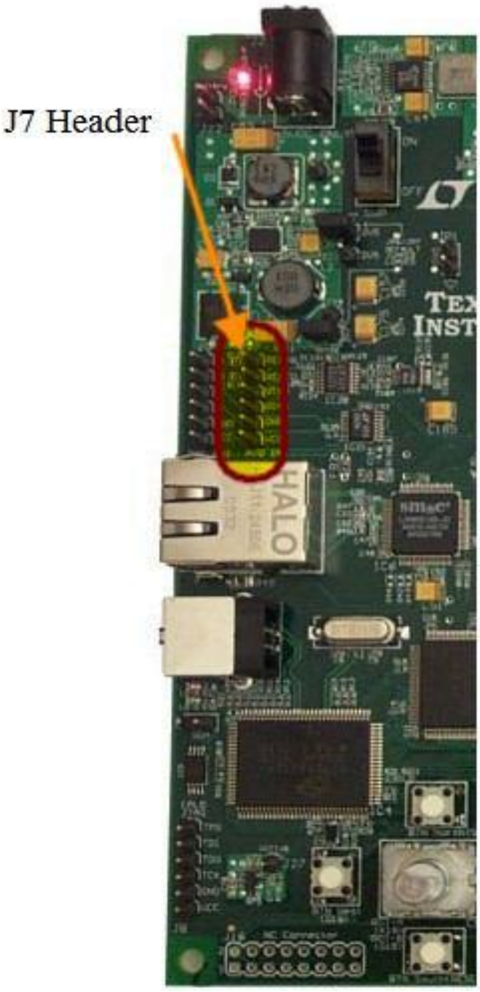


Figure 1: Two Channel Analog Capture Circuit as on the Spartan 3E FPGA board

The on-board ADC is Successive Approximation (SAR) type Analog to Digital Converter. It comprises the major constituents like a Digital to Analog converter, a comparator and a clock and a Successive Approximation register to store the digital data. DAC output is compared with the input voltage and the comparator gives a ‘0’ or ‘1’ as output. The block diagram of SAR -ADC can be shown as below:

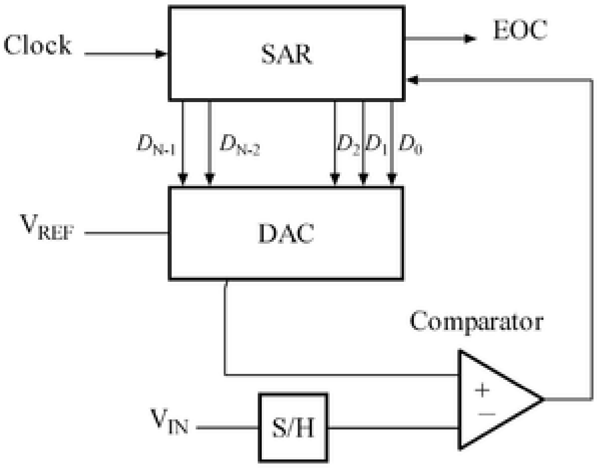


Figure 2: SAR- ADC.

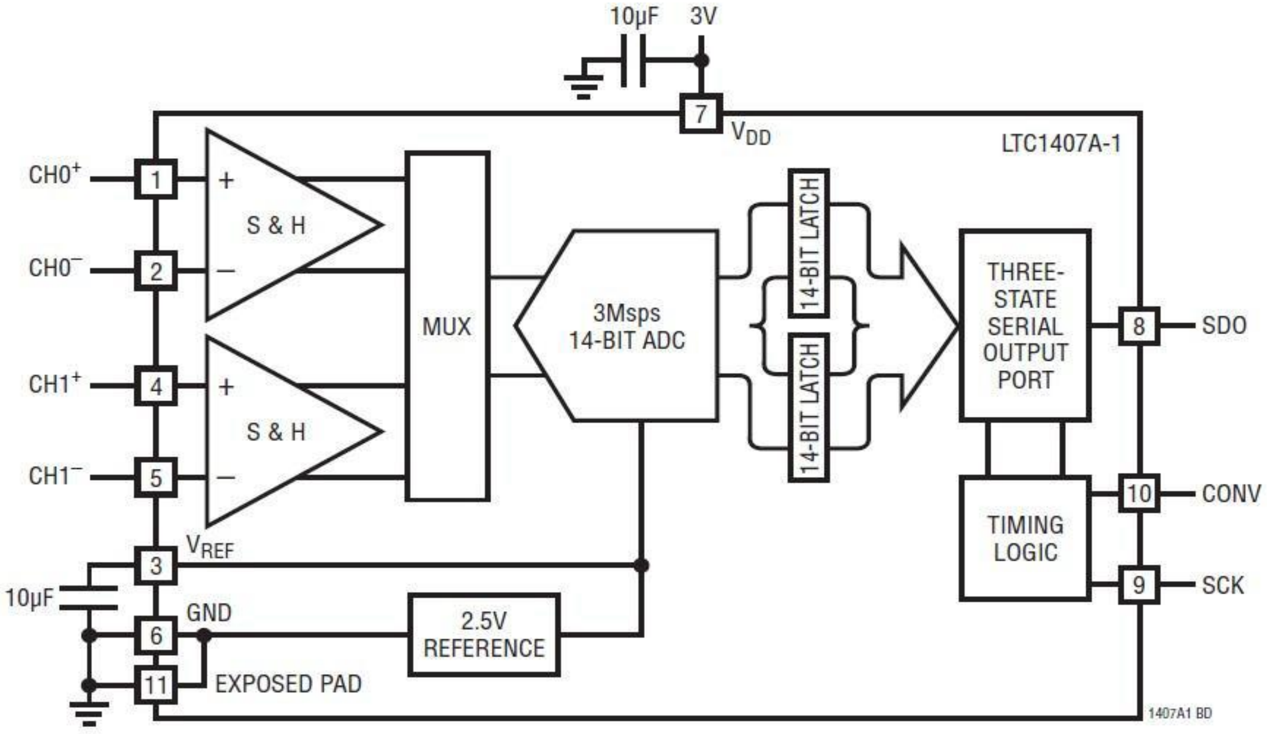


Figure 3: LTC1407A ADC

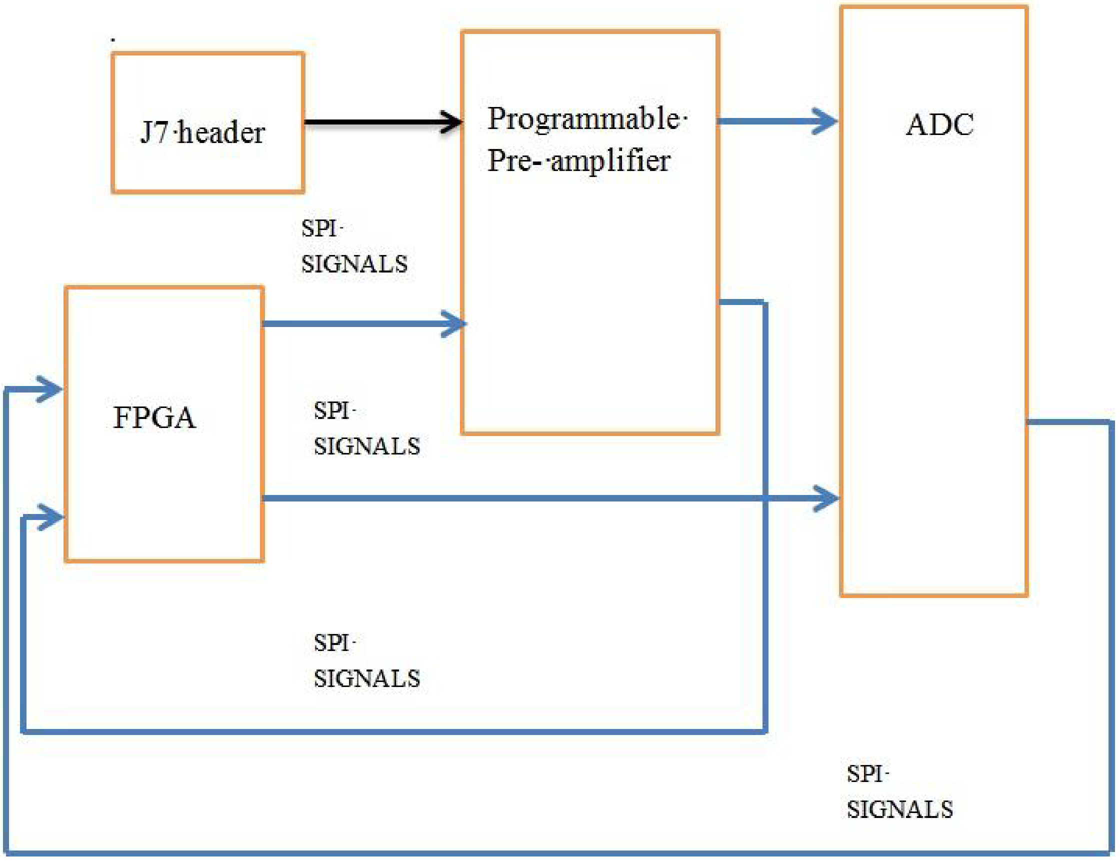


Figure 4: Analog capture circuit, featuring the input pins to be used and the ADC chip

2.2 **Programmable Pre-Amplifier**

There are two inverting amplifiers with programmable gain provided by LTC6912-1. The pre- amplifier scales the input voltage on channel A and B so as to expand the range of ADC i.e. 1.65 ±1.25V.

2.2.1 Features of Pre-Amplifier

- Channels with Independent Gain Control
- Channel-to-Channel Gain Matching of 0.1dB Max
- Offset Voltage of 2mV Max
- Gain Matching of 0.1dB maximum

- Extended Gain-Bandwidth at High Gains
- 3-Wire SPITM Interface
- Input Noise: 12.6nV/√ Hz
- Total System Range to 115dB

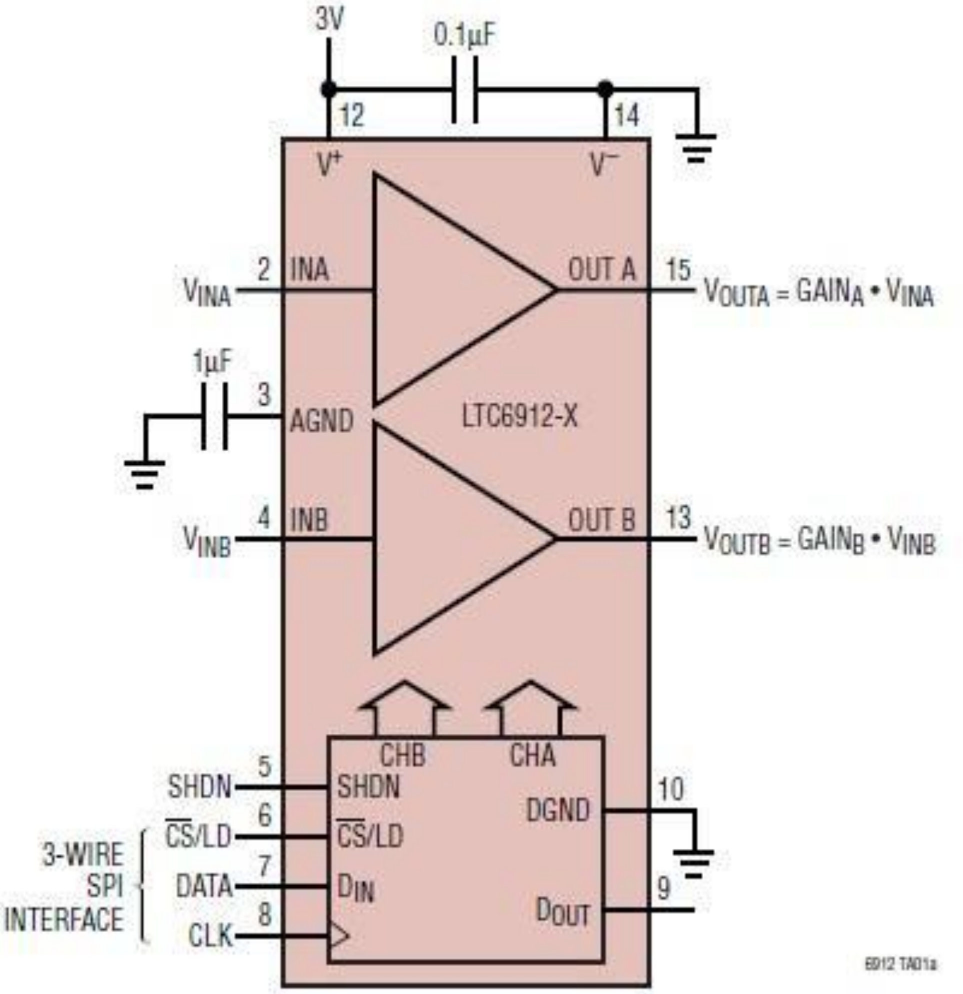


Figure 5: Low Noise PGA (16-Lead SSOP Package)

The LTC6912 is a family of dual channel gain amplifiers (PGA) which occupy very little space on board. Using a 3-wire SPI interface the gains can be programmed and set at 0, '- 1', '-2', '-5', '-10', '-20', '-50', and '-100'V/V (LTC6912-1).The LTC6912 family comprises two matched inverting amplifiers and they will operate on rail-to-rail input signals when operated with gain of unity. Single power supply applications are supported by a half-supply reference at AGND pin.

2.2.2 Communication between Pre-Amplifier and FPGA

Different interfacing signals are used by amplifier with the FPGA. SPI bus is used for the communication between amplifier and the FPGA. The SPI bus is also used by other devices like DAC, ADC, Platform Flash and StrataFlash. Hence the signals which are used by these devices in interfacing with the board must be disabled for proper working of ADC. These signals are listed below:

(a) SPI_MOSI: MOSI stands for Master Output, Slave Input. This signal is available at pin T4 on the board. It is sent from the FPGA to the ADC and presents the 8-bit gain settings according to which conversion takes place and range is determined.

(b) AMP_CS: It is the active low chip select signal available at the pin location N7 of the FPGA board. It has direction from the FPGA to the amplifier. The gain for amplifier is done when this signal becomes high.

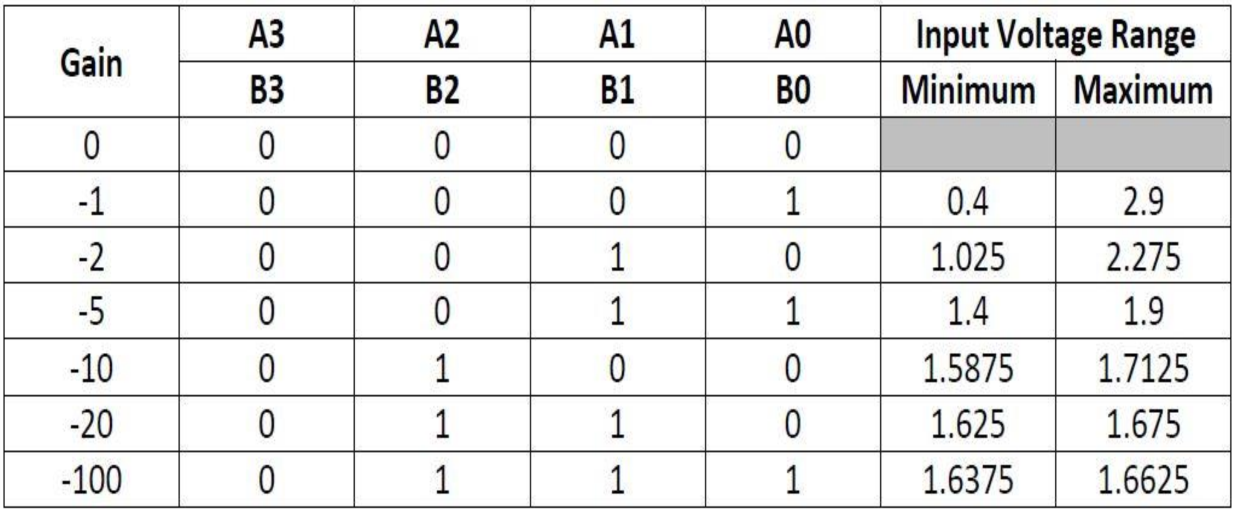
(c) SPI_SCK: This is the serial clock signal according to which conversion takes place. It is available at the pin location U16 of the board. Gain settings is done according to this signal. At the rising edge of the serial clock, SPI_SCK clock the SPI_MOSI signal is transmitted bit by bit. There is a signal called AMP_DOUT that resend the gain back to the FPGA on the trailing edge of the serial clock signal.

(d) AMP_SHDN: It is an Active high reset shutdown signal. It is available at the pin location P7 of the Spartan board. It is sent by the FPGA to the pre-amplifier.

(e) AMP_DOUT: It sends the settings of gain back to the FPGA and can be ignored usually.

2.2.3 Gain settings:

The gain of pre-amplifier is set by the SPI_MOSI signal. It is sent by the FPGA to the amplifier as a 8-bit word. The word is of 8 bits and it consists of two fields of 4 bits, first four bits are for channel A and latter 4 bits are for channel B. When the AMP_CS signal becomes low, bus transfer begins. On the rising edge of the serial clock signal, amplifier receives the data on SPI_MOSI signal and gain is set according to it. The following tabulation gives various gain settings for pre-amplifier with the input voltages:-



Gain	A3	A2	A1	A0	Input Voltage Range	
	B3	B2	B1	B0	Minimum	Maximum
0	0	0	0	0	▨	▨
-1	0	0	0	1	0.4	2.9
-2	0	0	1	0	1.025	2.275
-5	0	0	1	1	1.4	1.9
-10	0	1	0	0	1.5875	1.7125
-20	0	1	1	0	1.625	1.675
-100	0	1	1	1	1.6375	1.6625

Table 1: The allowable Gain settings and applied input voltage

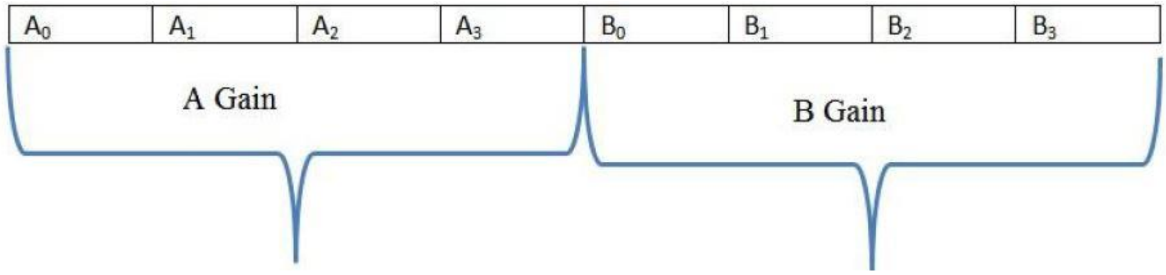


Figure 6: Gain register settings for channel A and B

2.3 Working of ADC (LTC 1407A)

The output of the ADC used is a 14-bit data in two's complement form. By setting the gain at '-1', the range is the ADC is set at 2.5 V. The formula according to which conversion is done is as below:

$$D[13:0] = GAIN \times \frac{V_{IN}-1.65V}{1.25V} \times 8192$$

The term D[13:0] is the digitized 14-bit data in two's complement form. GAIN is the settings given by the user to change the range of ADC. V_{IN} is the input voltage applied at channel A or B. 1.65 V is the reference voltage of ADC. 1.25 V is the medieval range of ADC so the output is scaled by this factor. Since the output is 14-bit, it is multiplied by two to the power 13 i.e. 8192.

2.3.1 Communication between FPGA and ADC:

(a) SPI_SCK: This signal is the clock which has an major role in the conversion process of analog to digital form. According to this the data is sent from the ADC chip to the FPGA.

(b) SPI_MISO: MISO stands for Master Input , Slave Output. This is the serial data output transmitted from the ADC chip to the FPGA board. It is gives the digital value of the applied analog input in the form of 14-bit two's complement binary value. This signal is generated internally available at pin N10 of the FPGA board and has direction from the ADC to the board

(c) AD_CONV: It is active high rest signal indicates the start of the conversion process of the analog signal. Since the signal is generated inside the FPGA, , it cannot be viewed with the help of oscilloscope. This signal is directed from the FPGA to the ADC and is available at pin P11.

2.4 **Methodology**

The methodology adopted in the interfacing of the on-board ADC of the Spartan 3E FPGA platform kit is as mentioned below:

(a) At first, the basic architecture and design of the on-board ADC chip is understood. A state diagram is drawn to implement the same via the FPGA starter kit.

(b) Then proper care is taken to keep adequate number of SPI_CLK clock cycles maintained while designing the state machine. It is needed for proper conversion process of analog to digital data.

(c) VHDL code is written to interface of the ADC with the real world signals keeping in view the various interface signals that are available in between the FPGA and the ADC. The main algorithm used in writing the program for the interfacing of the ADC is as mentioned below:

(i) The SPI_MOSI signal is transmitted bit by bit with the MSB sent first, so as to program the gain register with the required gain settings.

(ii) The analog to digital converter starts conversion of the given analog data after the gain is set and assertion of the high AD_CONV signal is done.

(iii) The total clock cycles that are required for the entire analog to digital conversion process are 34 clock cycles. The ADC enter into a tri-state before and after the digital conversion process .

(iv) After 34 clock cycles, AD-CONV signal goes high, the digital output is represented as SPI_MISO signal.

(v) The implementation constraints file (.ucf) is also coded for the above VHDL program with the pin locations of the signal sources being the same as mentioned in the user guide of the Spartan 3E FPGA kit. It is done as per the user guide as the signals are internal and the pins, at which those signals are available as output, are fixed while manufacturing the board itself.

(vi) Then synthesis of the VHDL code is done and the programming file is generated. It is now dumped into the FPGA kit via the platform cable USB with the usage of any Integrated Software Environment. The dumping of the programming file (‘bit’ file) would interface the on-board ADC of the FPGA kit so that the ADC could process the real world signals and give the digital representation of the given analog input.

(vii) Then the analog signal is given at the J7 header which is present for the interfacing of the real world signals in the FPGA kit. Then the results obtained are observed and verified. Two methods are used for displaying and verifying the output obtained from ADC. In one method, the on-board LEDs are used to verify the result bit by bit and in the second method, ChipScope Pro analyser is used to verify the results of the analog to digital conversion process. The major limitation in the case of showing the output via LEDs is the availability of fewer LEDs (8 on-board LEDs) than the required number of bits (14-bit digital output). As a result the significant 8 bits were displayed via the LEDs and the rest of the other lower bits were ignored. Then the program is again run, this time ignoring the 8 significant bits obtained earlier, the rest 6 bits are displayed via the LEDs so as to bring out a complete picture of the conversion process. In case of displaying the output via the ChipScope Pro, JTAG chain helps in getting the signals from the debug ports of the kit and displays the same via the ChipScope Pro analyser. The waveforms obtained via the ChipScope Pro are then analysed for verifying the results of the analog to digital conversion process.

2.5 **Results**

2.5.1 Showing output via LEDs

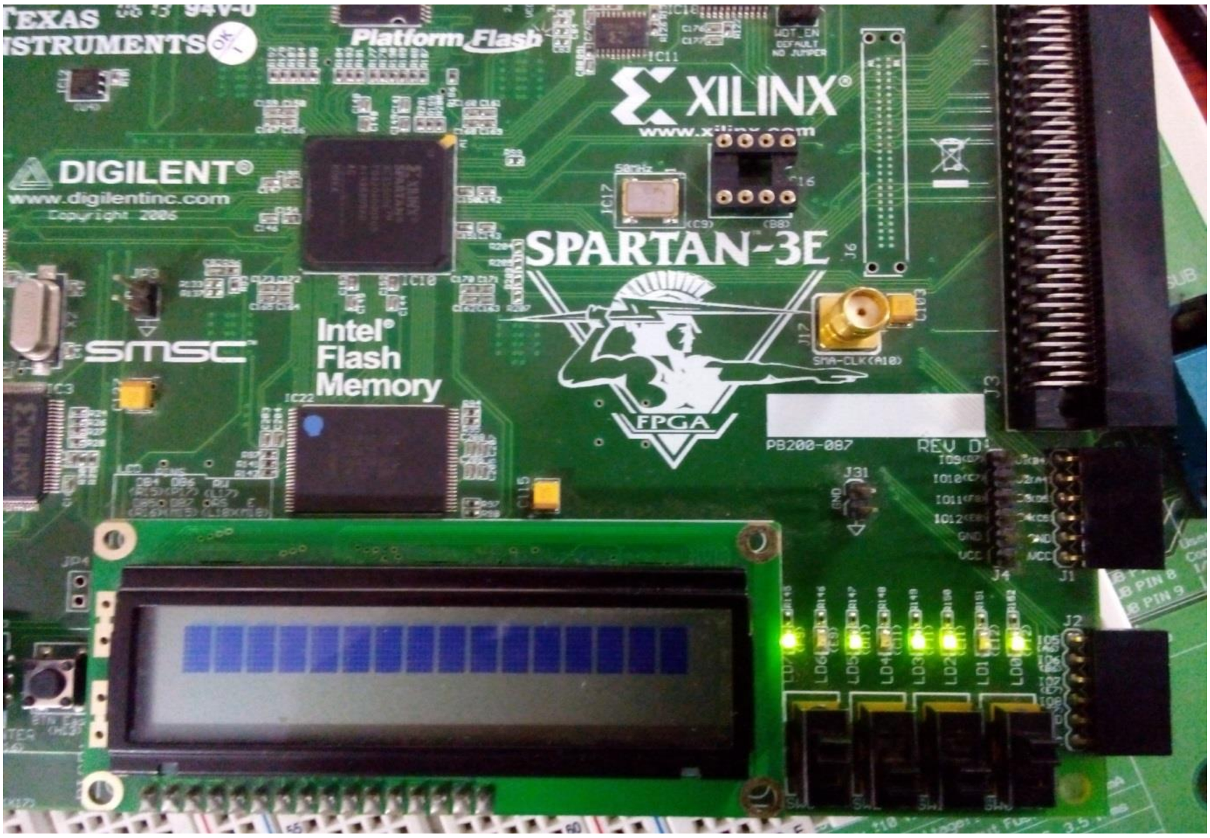
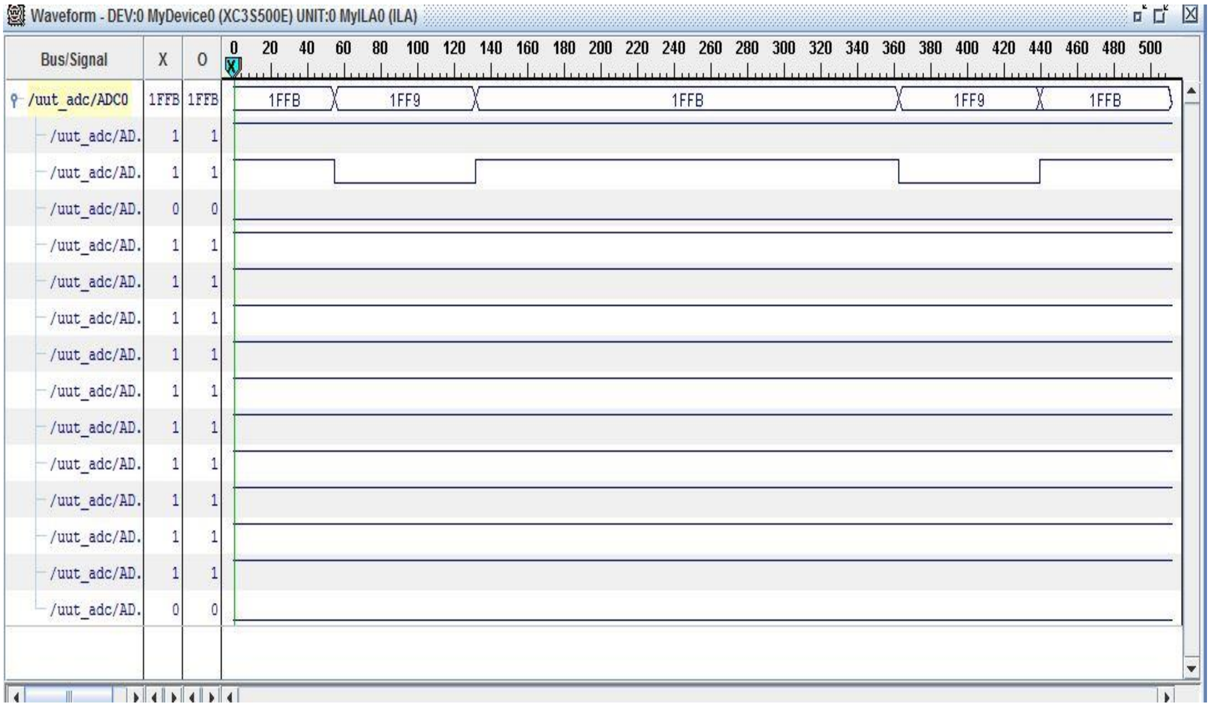


Figure 9

The main method was writing the VHDL program for the interfacing of the ADC with the real world signals and then interfacing the LEDs with the digital output of the ADC so that they would glow in accordance to the results output by the ADC. The ADC output was used to glow the on-board LEDs that are available in the Spartan 3E kit.

2.5.2 Showing the output via Chipscope Pro



Bus/Signal	X	O
/uut_adc/ADC0	1FFB	1FFB
/uut_adc/AD.	1	1
/uut_adc/AD.	1	1
/uut_adc/AD.	0	0
/uut_adc/AD.	1	1
/uut_adc/AD.	1	1
/uut_adc/AD.	1	1
/uut_adc/AD.	1	1
/uut_adc/AD.	1	1
/uut_adc/AD.	1	1
/uut_adc/AD.	1	1
/uut_adc/AD.	1	1
/uut_adc/AD.	1	1
/uut_adc/AD.	1	1
/uut_adc/AD.	0	0

Figure 8: ChipScope Pro output waveform on the application of 0.4 V DC supply voltage to the ADC.

CHAPTER 3:DISTRIBUTED ARITHMETIC METHOD OF FIR FILTER DESIGN

3.1 **INTRODUCTION**

In current scenario, there are immense developments in the implementation of Digital Signal Processing (DSP). Discrete Cosine Transform (DCT), Discrete Fourier transform (DFT) and Discrete Wavelet Transform (DWT) are some of the common algorithmic implementations. FPGAs have an edge over DSPs in the way that due to the sequential based architecture, the performance of DSPs are sometimes not up to the mark but FPGAs are efficient as they possess both sequential and concurrent architecture. Discrete Wavelet Transform (DWT) is used for denoising of signals, compression of images and in video processing. Filter banks are used by DWT especially in the case of designing FIR Filters. FIR filters are used widely due to their linear phase and stability. All frequency components in linear phase filters tend to have equal time delay since such filters have constant group delay. A filter having non-linear phase results in phase distortion.

The convolution operation is the underlying principle of FIR filter operation. FIR filters are implemented using several methods. These methods are analysed in terms of area, power consumption and rate of operation. The technique of Distributed Arithmetic is first proposed by Croisier and refined further by Peled and Liu. It is an efficient method which saves about 50 percent of hardware resources.

Linear Phase:-

Linear phase of a filter is defined if its impulse response has

Symmetric nature:

$$h(n) = h(p-n) \quad \text{for} \quad 0 \le n \le p$$

or representing it anti-symmetrically the equation of the response is:

$$h(n) = -h(p-n) \quad \text{for} \quad 0 \le n \le p.$$

We can define 4 types of FIR filters depending on even and odd sequence. These are as follows:

1. Antisymmetric filters, which may be of even or odd lengths.

2. Symmetric filters , which may be of even or odd lengths.

For implementing FIR filters, two methods can be used: (1) non-polyphase structure (2) polyphase structure .The latter implementation breaks the FIR filter impulse response into N different small filters, where N is the decimation factor. The polyphase filtering multiplies specific input values with selected values of the impulse response. For instance when N = 2, x(0), x(2), x(4), . . . which are the input values combine only with the filter coefficients h(0), h(2), h(4), . . ., and the input values x(1), x(3), x(5), . . are multiplied with the coefficients of the filter h(1), h(3), h(5), . . .

Polyphase configuration

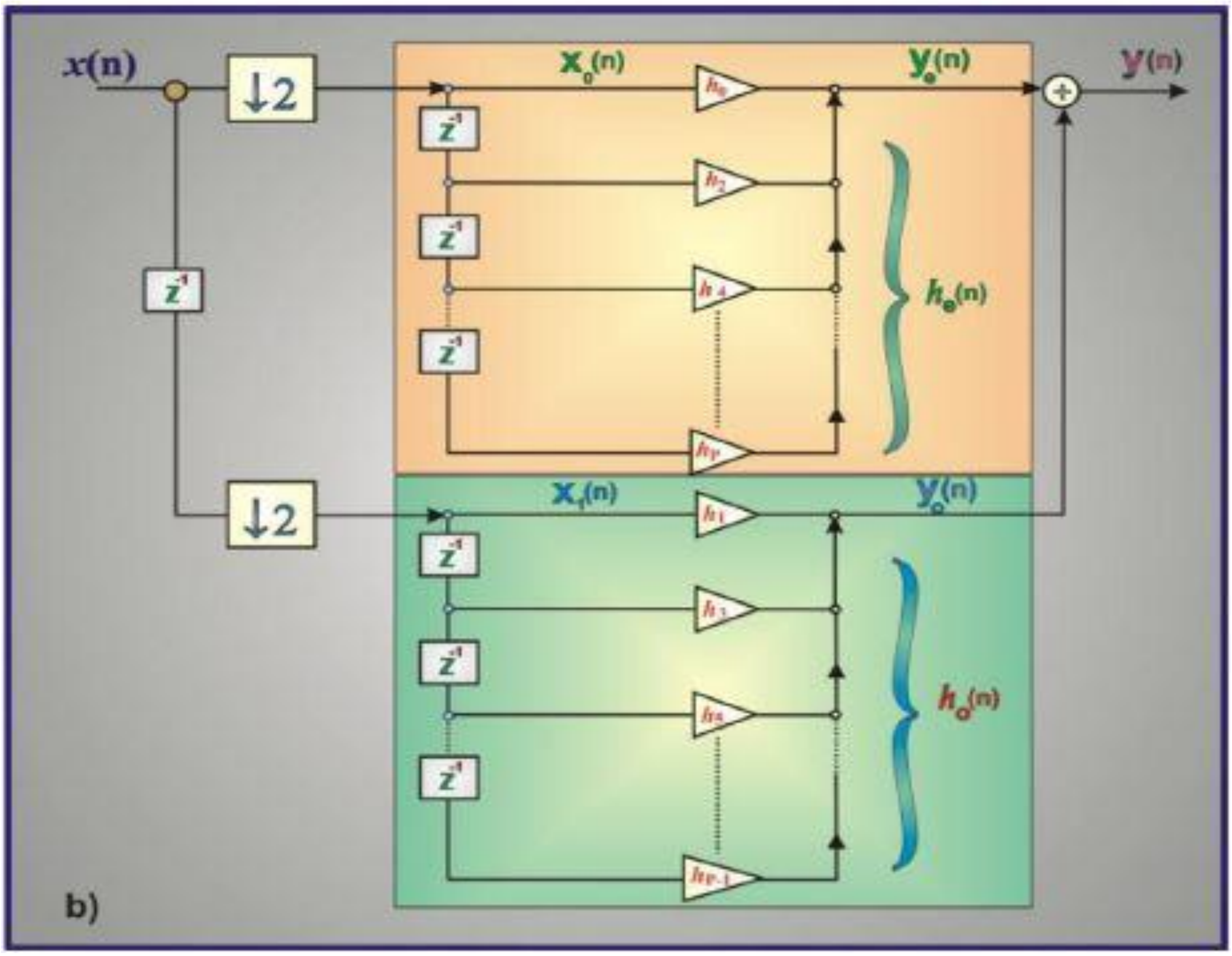


Figure 11

In the non-polyphase structure, decimation or down sampling is done after filtering. The filter computed many samples of which half were discarded. Due to this, the design is not efficient.. In contrast to this, the polyphase structure first does the down sampling and then breaks the sample into even and odd.

Non-polyphase configuration

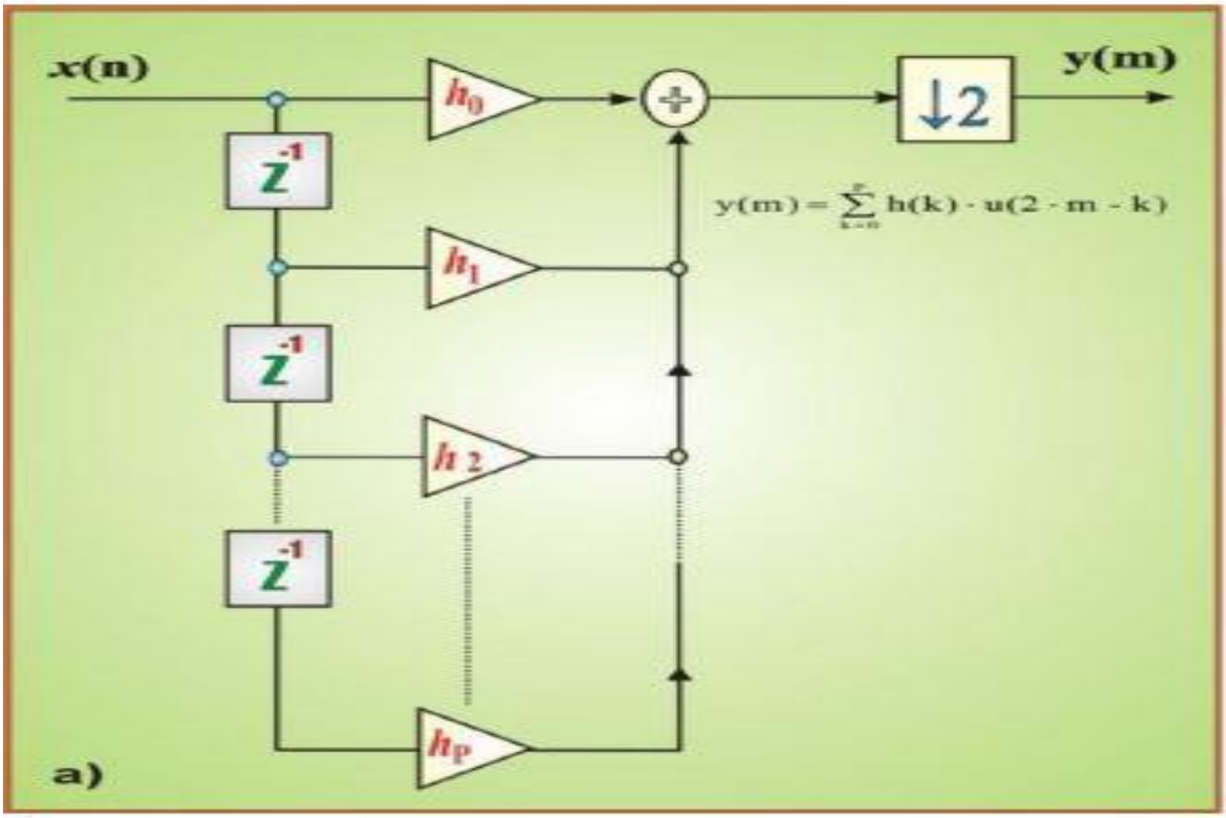


Figure 12

DISTRIBUTED ARITHMETIC

In order to reduce hardware cost, all the bits of an input data can be processed in a parallel manner using Look Up tables. But in Distributed Arithmetic structure, bitwise operation is performed. The input data is processed bit by bit, first processing the least significant bit and then rest of the bits. There is substantial reduction in hardware as all bits have to pass through same architecture. In other words, if there is a N-bit parallel design, the DA method requires only (1/N)th of the hardware resources. As a result only one clock cycle is required for execution while about N cycles are required in serial execution. However, for the serial design the time-hardware product is smaller than the parallel design because the propagation delays are generally smaller as compared to the parallel structure.

There are four building blocks in this method which are as follows:

(a) A splitter

(b) Data Register

(c) ROM

(d) A controller

The Splitter

This constituent does the polyphase configuration of the filter. The data sequence is separated into even and odd samples and these are then processed in parallel fashion. The input data is read according to a “load signal” and transferred in alternate fashion to the odd and the even output as directed by the load signal. The splitter has two states and the frequency of the load signal is linked to the time taken in the processing of a sample.

Data Register

By parallel shifting operation, even samples are stored in registers. After the ((P/2) + 1) [(P + 1)/2] even samples are read, the first and last even samples are added while the second and next to the last one even samples are added. Such process is carried out with all the even samples. In the filter is of even order case, the central one is kept as it is since it represents the median of the input data. The output is around ((P + 2)/4) [round ((P + 1)/4)] parallel data which constitute an even address conveyed by round ((P + 2)/4)[round((P + 1)/4)] bits, allowing the access of the evenlook-up table. Similar procedure is adopted for odd samples. The number of odd samples are (P/2) [(P+1)/2] and the process of loading, shifting and adding gives round P/4)[round((p + 1)/4)] bits making the odd address of LUT.

The ROM or LUT design

During implementation in FPGA, the area utilized in construction of the LUT which is decided by the length of filter is an important parameter. The size of the LUT used has been significantly reduced by using Distributed Arithmetic method, symmetry of the filter and its polyphase structure. Earlier the number of entries in the even LUT were $2^{((P/2)+1)}$ for even filter order and [$2^{(P+1)/2}$] for odd filter order, ‘P’ represents order of filter. Similarly for odd LUT the number of entries were $2^{(P/2)}$[$2^{(P+1)/2}$] entries. The figure inside the brackets gives the number of entries when filter order is odd. By using this method the addresses in LUT are reduced from $2^{((P/2)+1)}$ [$2^{(P+1)/2}$] to 2($^{\text{round}((P+2)/4))}$ [$2^{(\text{round}((P+1)/4))}$] for the even section and from $2^{(P/2)}$ [$2^{((P+1)/2)}$] to $2^{(\text{round}(P/4))}$ [$2^{(\text{round}((P+1)/4))}$] for the odd section. The reduction   (or gain) is by a factor of 4. Consequently less power is consumed by smaller ROMs. The floating point implementation of the FIR filter coefficients consume a lot of power and hardware resources, thus leading to low operating speed of circuit. To overcome this problem, the filter coefficients are multiplied by a factor of 2^R where R represents the resolution of the input samples chosen by the user. It is stored in fixed-point two's complement form. In Figure 14, k represents the address pointer of the LUT. It is composed of 3 bits for even LUT and 2 bits for odd LUT . The 2 bits addresses are extracted from S4 and S3and the 3 bit addresses are obtained from S0, S1 and S2.

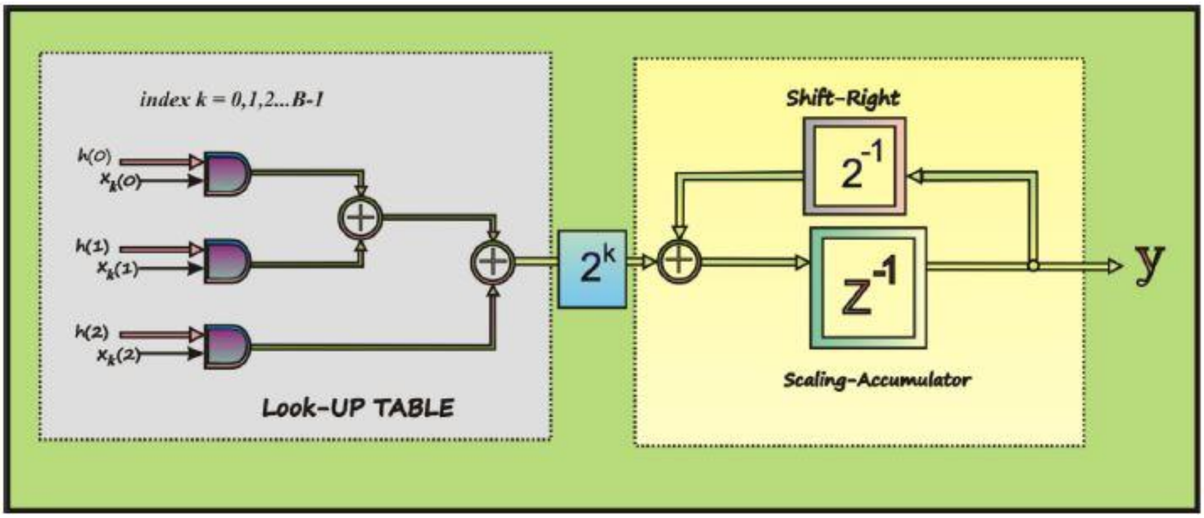
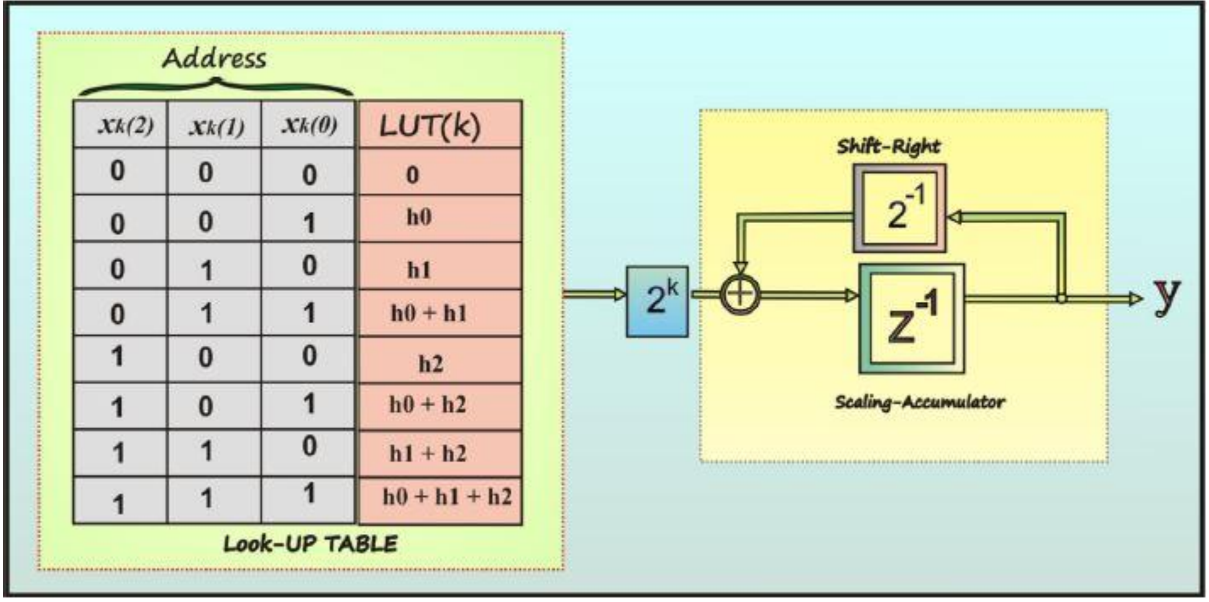


Figure 13



$x_k(2)$	$x_k(1)$	$x_k(0)$	LUT(k)
0	0	0	0
0	0	1	h0
0	1	0	h1
0	1	1	h0 + h1
1	0	0	h2
1	0	1	h0 + h2
1	1	0	h1 + h2
1	1	1	h0 + h1 + h2

Figure 14

The Controller

The controller comprises a accumulator performing scaling and time controller. The recursive equation is implemented by the scaling accumulator. The generation of the sampling "load" signal for controlling the splitter and assertion of the signal is performed by the controller. The result is to right R times to reverse the scaling and to get the accurate data .

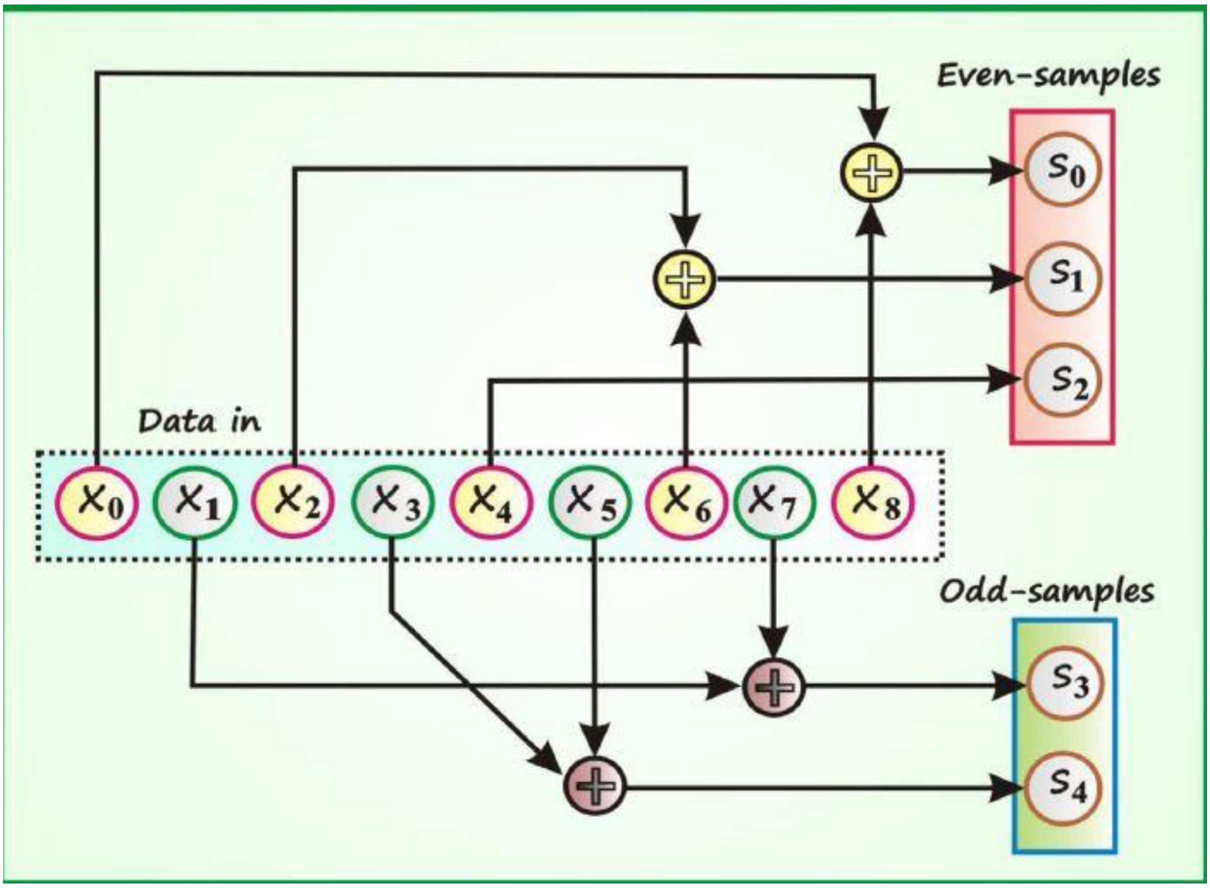
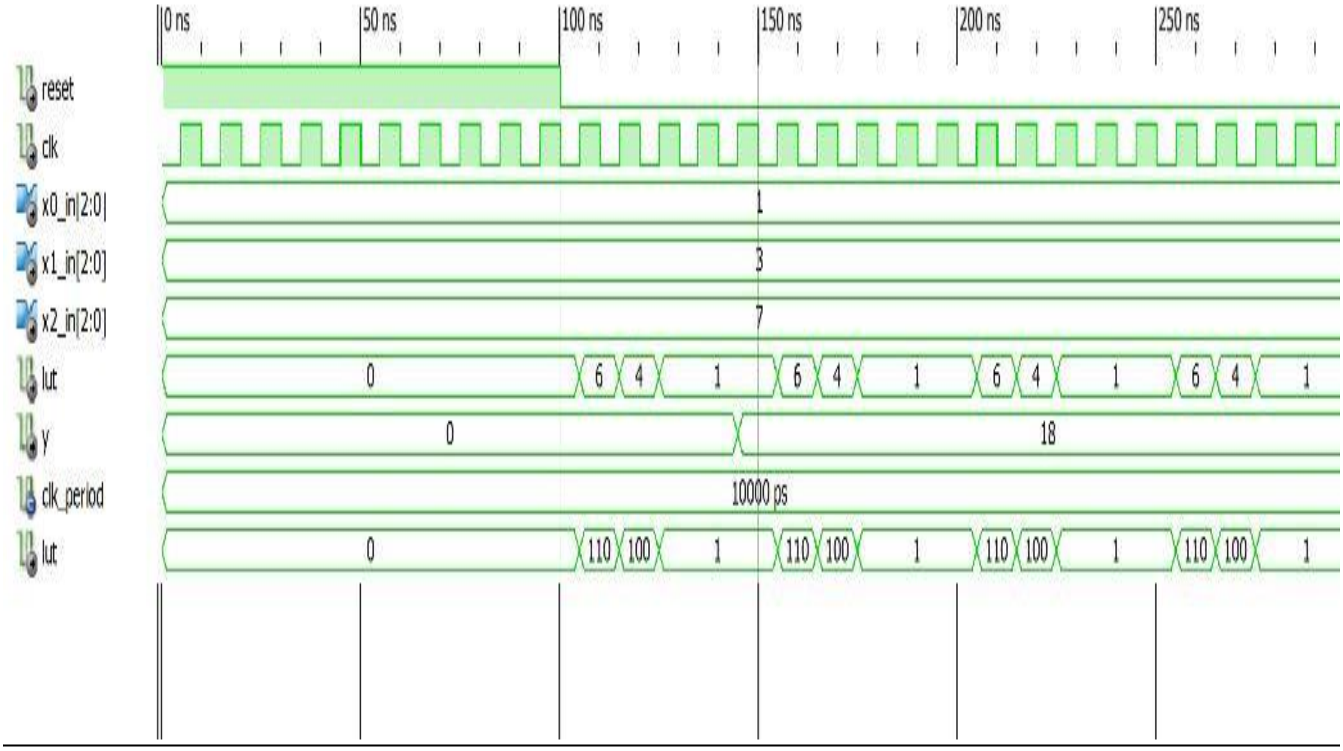


Figure 15

The LUT or ROM design is simulated in XILINX and the result is obtained successfully.

3.2 **SIMULATION RESULTS**



The inputs taken were

X0=(1)$_{10}$=(001)$_{2}$
X1=(3)$_{10}$ =(011)$_{2}$
X2=(7)$_{10}$ =(111)$_{2}$

and the output obtained is

y= (18)$_{10}$

CHAPTER 3:

DAC INTERFACING

3.1 **INTRODUCTION**

The Spartan-3E FPGA board contains a four-channel, serial Digital-to-Analog Converter (DAC) compatible with SPI bus. The DAC LTC2624 is Linear Technology quad DAC possessing 12-bit unsigned resolution for data. J5 header contains pins on which input can be provided to DAC .

Features of DAC:-

- Quadrature type 12 bit DAC. It has 4 independent DACs within it which we can use one or all of them at a time.
- Supply Range extends from 2.5 V to 5.5 V. It can accept its power applied in between 2.5 to 5.5 Volts but anything above 5.5 V is not acceptable.
- Operation at low power: 250µA per DAC at 3V. It is consumes less power and so it is ideal for applications at chip level. The current is low (1 microampere maximum) that passes through DAC. Hence there are less chances of short circuit or overheat.
- Double Buffered Digital Inputs. The incoming inputs are twice relayed and then sent to the DAC It implies that it acts like a pair of master and slave.
- High Rail-to-Rail Output Drive (±15mA). It has high output swing reject any of the incoming input signals and that while conversion no bit is missed.
- Very low Crosstalk between DACs (about 5µV or less). It ensures that all 4 DACs can work together without interfering with each other.
- Power-On Reset to Midscale. It means the DACs can be set to their midscale values without restarting the entire process at any stage.
- 16-Lead Narrow SSOP Package. It has compact and narrow package which makes it portable and hence helpful everywhere .

3.2 APPLICATIONS OF DAC

- In automatic testing of devices: For automatic testing pf any equipment, an analog component is required for its proper functioning. Sometimes the signals are sent from the control room also. Hence it is essential to convert digital to analog signal. Here DAC comes into picture.

- In Process Control and Automation in industries: For industrial automation purposes, digital signals are required. But analog signals are good for transmitting signals over long distances. So there is a need of converting digital data to analog.

- In mobile communications: Analog signals are suitable for transmitting over long distances. But digital data are easier and well suited for various operations and processing in mobile communications. Hence digital data need to be converted to analog after processing.

3.3 Working of DAC

Reference voltage for DAC at channel A and B is 3.3V whereas that at channel C and D is 2.5V.There will be slight corresponding variances in the output voltage since the reference voltages have a tolerance of ±5%tolerance .The conversion of digital to analog data by DAC takes place by following equation:

$$V_{OUT} = \frac{D[11:0]}{4096} \times V_{REFERENCE}$$

where $V_{REFERENCE}$ = 3.3V for channels A and B and 2.5V for channels C and D.

Since 12 bit unsigned number is converted into corresponding analog value by DAC , so the output is scaled by 2^{12} or 4096. And the result thus obtained is multiplied by $V_{REFERENCE}$ to get the desired analog output. All communication is done through SPI bus.
The characteristics of the SPI bus which make the bus unique for communication are:1. It is full duplex, 2. It is synchronous and 3. It is character oriented bus channel having four-wire interface. The serial data (SPI_MOSI) is transmitted by the bus master to the selected bus slave and this selection is done through address of the slave device. It drives the bus clock signal (SPI_SCK) . The serial data (SPI_MISO) is echoed back to the bus master by the slave simultaneously but this signal can be ignored.

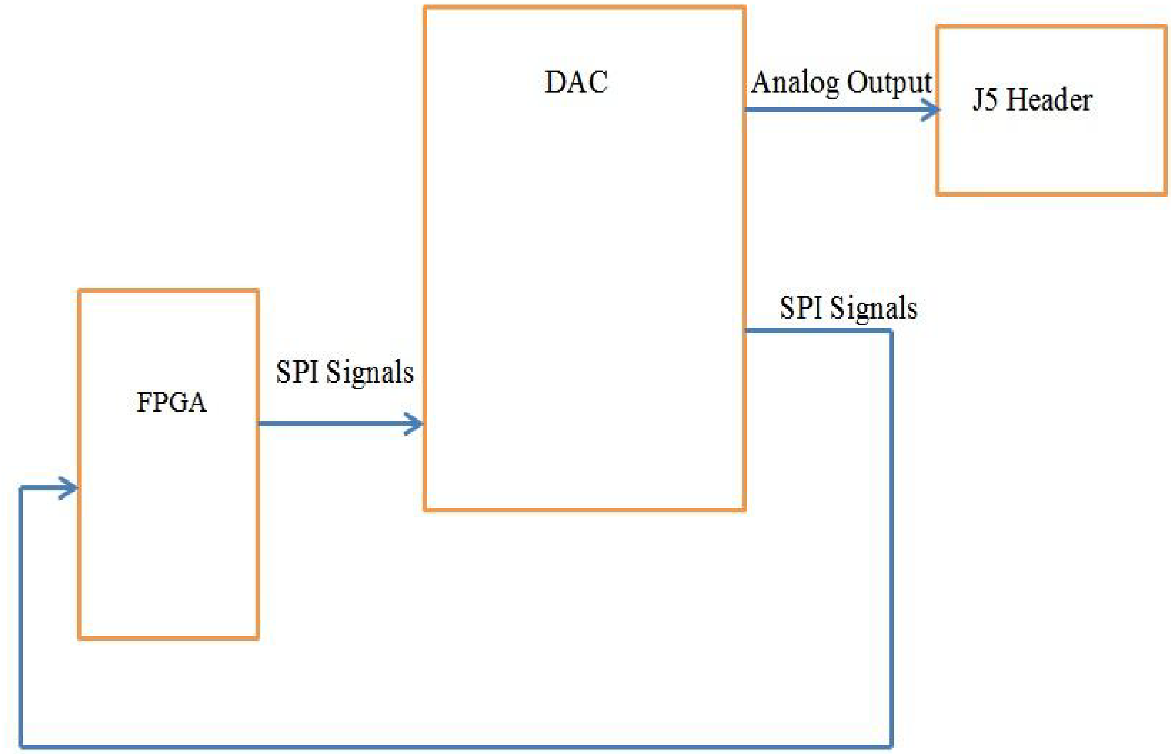


Figure 16: Block diagram for the Digital to Analog convertor connection schematics

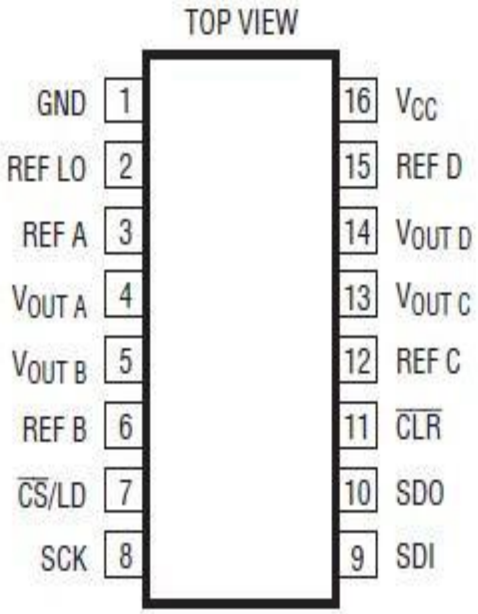


Figure 17: The Top view of the LTC 2624 DAC chip.

3.4 **Interface signals between the FPGA and the DAC**

Following are the interface signals used to interface the DAC with FPGA :-

(a) DAC_CS: This signal is an active low chip select signal.When this signal returns high, the conversion starts.This signal has direction from the FPGA to the ADC.

(b) SPI_MOSI: It is the 12 bit unsigned serial data directed from FPGA to DAC. It is converted from digital to analog form by the Digital to Analog converter. It is available at pin T4 of FPGA board.

(c) SPI_SCK: This is the bus clock signal according to which conversion process takes place.

(d) SPI_MISO: It is the serial data that is directed from DAC(slave) to the FPGA(master) and represents the analog output.

(e) DAC_CLR: This signal is the asynchronous, active low reset input given to the DAC to reset the settings.

3.5 **Communication Protocol for DAC**

There are two protocols supported by DAC through the SPI bus which is described as below:-

- 24-bit protocol

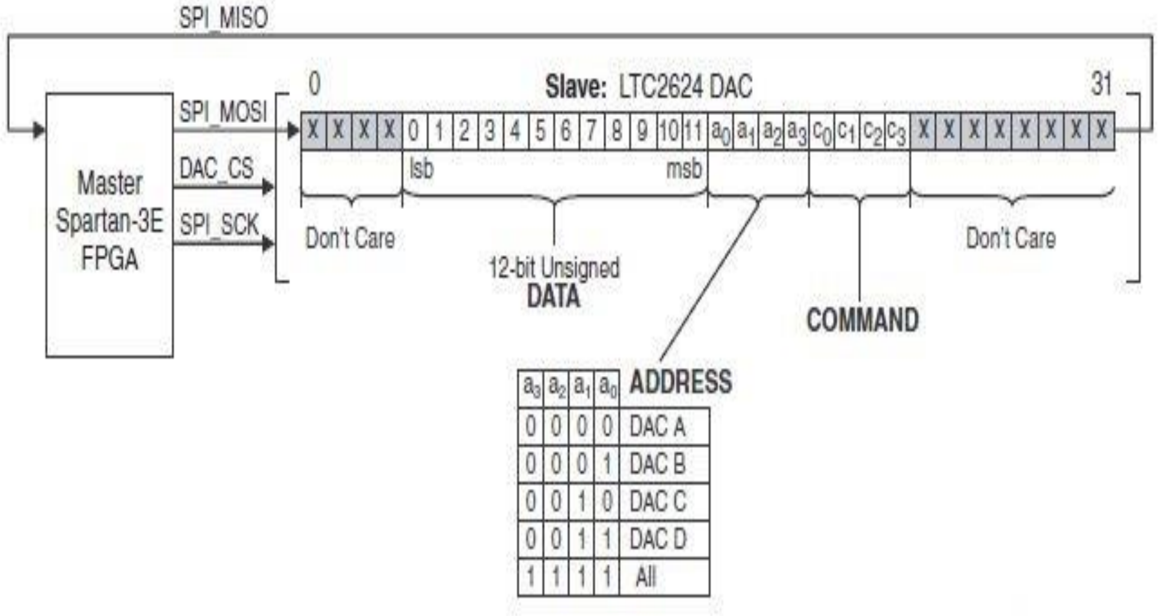
– When the CS/LD input is taken low, it behaves as chip-select signal, enabling the SDI and SCK buffers and also the input shift register.

– First the 4-bit command is loaded, then the 4-bit address, A3-A0 and then 16-bit data word. The data word represents the input word, Then 4 don’t care bits are sent.

–Only when CS/LD is low, data can be transferred. The rising edge at CS/LD causes the end of the data transfer.

• 32-bit protocol

--In this protocol, at first the FPGA transmits eight ‘Don’t care’ bits and then the same procedure is followed as the 24-bit protocol.



a_3	a_2	a_1	a_0	ADDRESS
0	0	0	0	DAC A
0	0	0	1	DAC B
0	0	1	0	DAC C
0	0	1	1	DAC D
1	1	1	1	All

Figure 18: SPI Communication Protocol

3.6 **Methodology**

To interface the on-board DAC, the methodology used is as follows:

(i) The program for DAC is written in VHDL language and the 12 bit digital data is input in the program to the master device.

(ii) This 12 bit data is given as an output to the DAC to convert the data into analog value by the DAC .

(iii) Through VHDL program, 4-bit command word and 4-bit address are given for selection of the desired DAC .

(iv) The input 12 bit data is processed by DAC. According to the reference voltage (which again depends on the output DAC chosen) the conversion of the digital data into analog value takes place which is observed through Digital Storage Oscilloscope (DSO).

(v) A ramp signal is to be generated as the output for an inceasing pattern of input.

3.7 **Results**

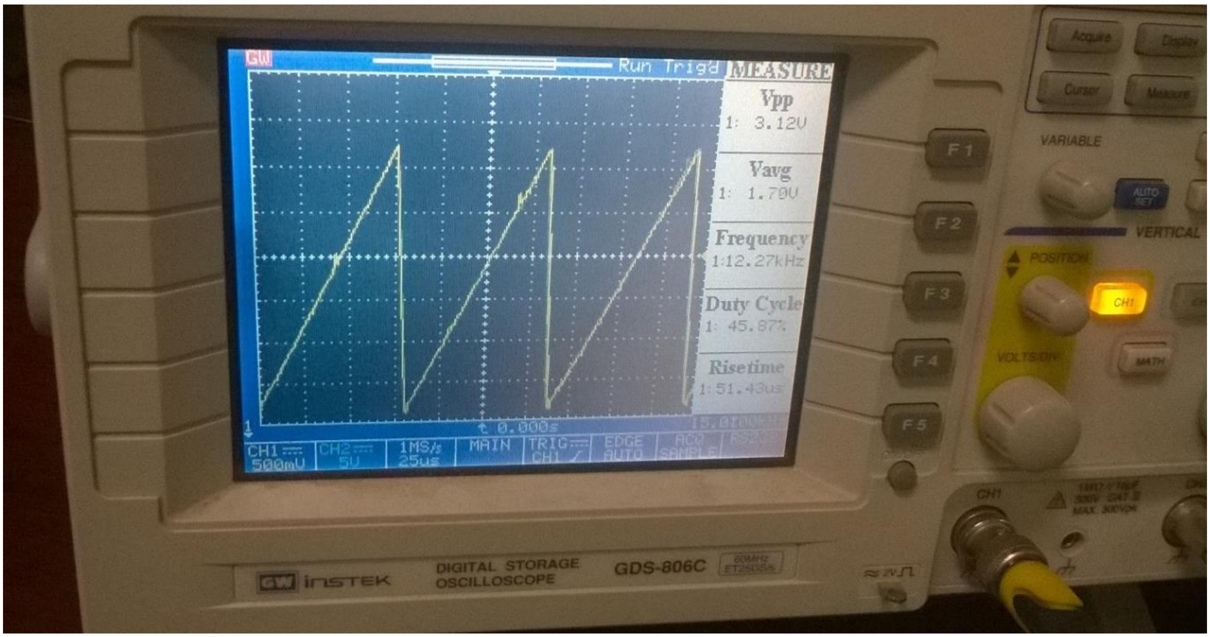


Figure 19: Showing ramp signal generated in DSO

After analysing the ramp signal, it is observed that output obtained at the chosen DAC (DAC of channel A) is a good approximation of the analog signal.

The problem encountered while using on-board ADC and DAC together in the Spartan 3E FPGA kit is that though the ADC and DAC have different resolution. ADC converts the analog value into a 14 bit digital data whereas only 12 bits are converted to the corresponding analog value by DAC. It causes loss of lower two bits of data. A separate DAC can be designed to eliminate the above problem but in that casr an independent system cannot be designed.

CHAPTER 4: CONCLUSION

- The on-board ADC & DAC of the Spartan 3e Board are properly interfaced with real signals.

- ChipScope-Pro is used to analyse the internally generated signals. Because of the low power consuming capability of the ADC it can handle any kind of data.

- The ADC is analysed for constant voltage supply

- The DAC is analysed and checked with a constant input.

- Mathematically it is proved that the ADC output observed through ChipScope-pro analyser and the value theoretically found are almost equal.

CHAPTER 5:

REFERENCES

[1] Spartan 3e FPGA Starter Kit Board User Guide.

[2] Mrs. Vidya H. Deshmuk Dr. Abhilash Mishra, Prof. A.S. Bhalachandra,“FIR Filter Design On Chip using VHDL” Volume 2, Issue 7, July 2014, International Journal of Computer Science.

[3] Gurneet Kaur , Amandeep Singh Sappal, “Design of FIR Filter using Distributed Arithmetic Architecture” ,2013, International Journal on Recent and Innovation Trends in Computing and Communication.

[4] U.Mayer- Baese, “Digital Signal Processing using Field Programmable Gate Arrays”, 3rd edition, Springer series, 2007.

[5] E. Christen and K. Bakalar. "VHDL-AMS-A Hardware Description Language for Analog and Mixed-Signal Applications". *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, Page no: 1263-1272, Oct. 1999.

[6] Prabhat Ranjan “Implementation of FIR Filter on FPGA” Department of Electronics and Communication Thapar University, Punjab (July 2008).

[7] M. Yamada, and A. Nishihara, “High-Speed FIR Digital Filter with CSD Coefficients Designed on FPGA”, in Proceedings of IEEE Design. Automation Conference , 2001, pp. 7-8.