Audio Visualiser using Field Programmable Gate Array(FPGA)

June 21, 2014

Aditya Agarwal
Computer Science and Engineering,IIT Kanpur
Bhushan Laxman Sahare
Department of Electrical Engineering,IIT Kanpur
Harsh Agarwal
Department of Chemical Engineering,IIT Kanpur
Sudhanshu Tandon
Department of Mechanical Engineering,IIT Kanpur

1 Abstract

Our aim is to represent the wave form of the input analog signal,using FPGA and analog to digital converter (ADC), on an LED matrix. With the help of ADC the input analog signals will be converted to digital and using Fourier Transforms and FPGA signal processing,the output will be displayed on an LED matrix.

2 Introduction

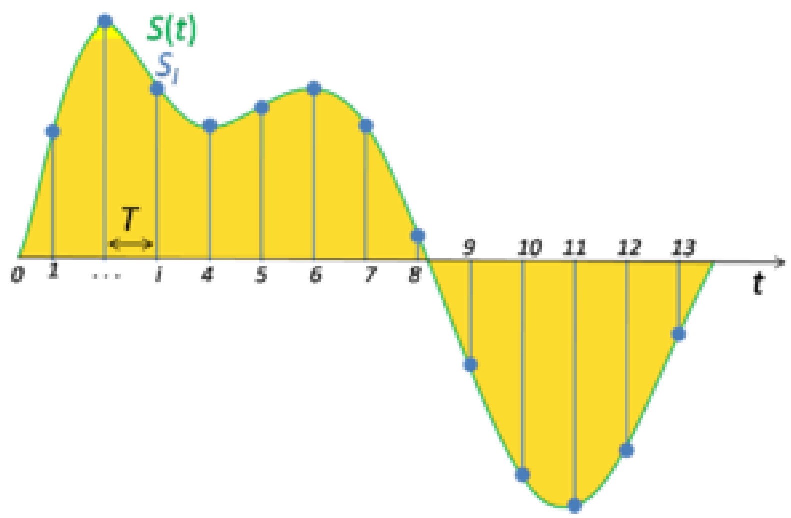
Signal processing finds its application in many areas as in image processing or audio processing.Real time processing of continuous signal can be done using FPGA.
A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer.The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC).Contemporary FPGAs have large resources of logic gates and RAM blocks to implement complex digital computations.
Digital circuits may become too big to be handled manually therefore FPGA provides a way to see the efficiency and working of such circuits without making them on breadboard but by manually writing the code in any of the HDL's like Verilog and by synthesising and stimulating it.
The Fourier transform is a mathematical transformation employed to transform signals between time (or spatial) domain and frequency domain, which has many applications in physics and engineering.It is reversible, being able to transform from either domain to the other.In the case of a periodic function over time (for example, a continuous but not necessarily sinusoidal musical sound), the Fourier transform can be simplified to the calculation of a discrete set of complex amplitudes, called Fourier series coefficients. They represent the frequency spectrum of the original time-domain signal.

3 Motivation

All the complex integrated circuits, before being hardwired and commercially produced, are tested on FPGA board. The circuit of those ICs are firstly coded using a hardware description language and is then synthesized. These synthesizable codes are then implemented on FPGA so that the circuit and its efficiency could be tested and improved.
This motivated us to learn about the HDLs and when we came to know that club is offering a project based on FPGA in summers,we decided to take it. After a lot of discussion with the coordinators,we finalised to make a audio visualiser using FPGA. FPGA provides a very efficient way to design very complicated digital circuits quite easily on boards which is great fun.

4 Background Theory

In signal processing, sampling is the reduction of a continuous signal to a discrete signal. A common example is the conversion of a sound wave (a continuous signal) to a sequence of samples (a discrete-time signal).



The sampling frequency or sampling rate, fs, is defined as the number of samples obtained in one second (samples per second), thus fs = 1/T.In practice, the continuous signal is sampled using an analog-to-digital converter (ADC).
Digital audio uses pulse-code modulation and digital signals for sound reproduction. This includes analog-to-digital conversion (ADC), digital-to-analog conversion (DAC), storage, and transmission. In effect, the system commonly referred to as digital is in fact a discrete-time, discrete-level analog of a previous electrical analog.
The discrete Fourier transform (DFT) converts a finite list of equally spaced samples of a function into the list of coefficients of a finite combination of complex sinusoids, ordered by their frequencies, that has those same sample values. It can be said to convert the sampled function from its original domain (often time or position along a line) to the frequency domain.
The input samples are complex numbers (in practice, usually real numbers), and the output coefficients are complex as well. The frequencies of the output sinusoids are integer multiples of a fundamental frequency, whose corresponding period is the length of the sampling interval.
A fast Fourier transform (FFT) is an algorithm to compute the discrete Fourier transform (DFT) and its inverse. Fourier analysis converts time (or space) to frequency and vice versa; an FFT

rapidly computes such transformations by factorizing the DFT matrix into a product of sparse (mostly zero) factors.

5 Implementation Details

5.1 Hardware and Software Involved

1. **Mojo FPGA board**
Specifications of the MOJO board:
(i) Spartan 6 XC6SLX9 FPGA
(ii) 84 digital IO pins
(iii) 8 analog input pins
(iv) 8 general purpose LEDs
(v) 1 reset button
(vi) 1 LED to show when the FPGA is correctly configured
(vii) Min Voltage: 4.8V
(viii) Max Voltage: 12V
(ix) A micro-controller (ATmega32U4) used for configuring the board and USB communication
(x) On board Flash memory to store the FPGA configuration file

2. **Two 8*8 LED Matrix**
Typically, these type of matrices have 16 pins to take the input, with different pins corresponding to different rows and columns. Specifications of the LED matrix:

Parameter	Maximum	Units
Power Dissipation per dot	65	mW
Peak Forward Current per dot	100	mA
Forward Current per dot	25	mA
Reverse Voltage	5	V
Operating Temperatur Range	-40 to 80	C
Storage Temperature Range	-40 to 85	C
Soldering Temperature	260 degrees for 5 seconds	

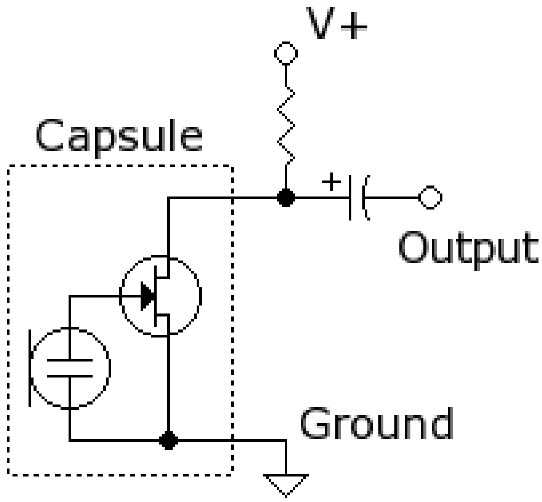
3. **Operational Amplifier LM324n**
An operational amplifier (op-amp) is a DC-coupled high-gain electronic voltage amplifier with a differential input and, usually, a single-ended output. It produces output voltage that is typically hundred and thousand times the input voltage. Specifications of the op-amp LM324n:
(i) Supply voltage: 32 volts DC voltage
(ii) Input voltage: -0.3 to 32V DC voltage
(iii) Input current: 50 mA
(iv) Maximum Power dissipation: 1420 mW
(v) Operating temperature range: 0 to +70 oC

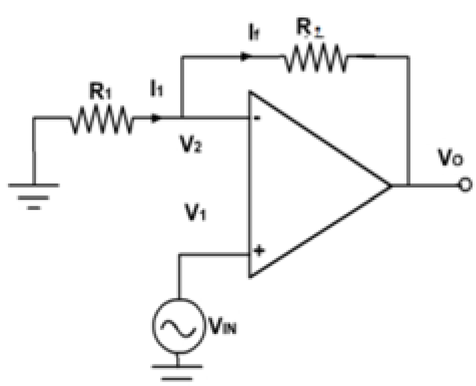
4. **Electret Microphone**
It is a type of Electrostatic Induction based microphone, which eliminates the need for a polarizing power supply by using permanently charged material.
(i) Sensitivity: -42 to -46 dB
(ii) Operating Voltage: Standard- 3V DC Max.- 10V DC
(iii) Output impedance: 2.2 kilo-ohm
(iv) Operating frequency: 20   20000 Hz
(v) Signal to noise ratio: 60 dBA
(vi) Operating temperature: -20   +70 oC

5.**Capacitors and Resistors**
(i) 0.1 micro-farad
(ii) 22 ohm
(iii) 500 ohm
(iv) 22000 ohm
6. **ISE Design Suite 14.7**
This software is used to write code in Verilog and stimulate and synthesise them and then upload them on FPGA.

5.2 Our Approach

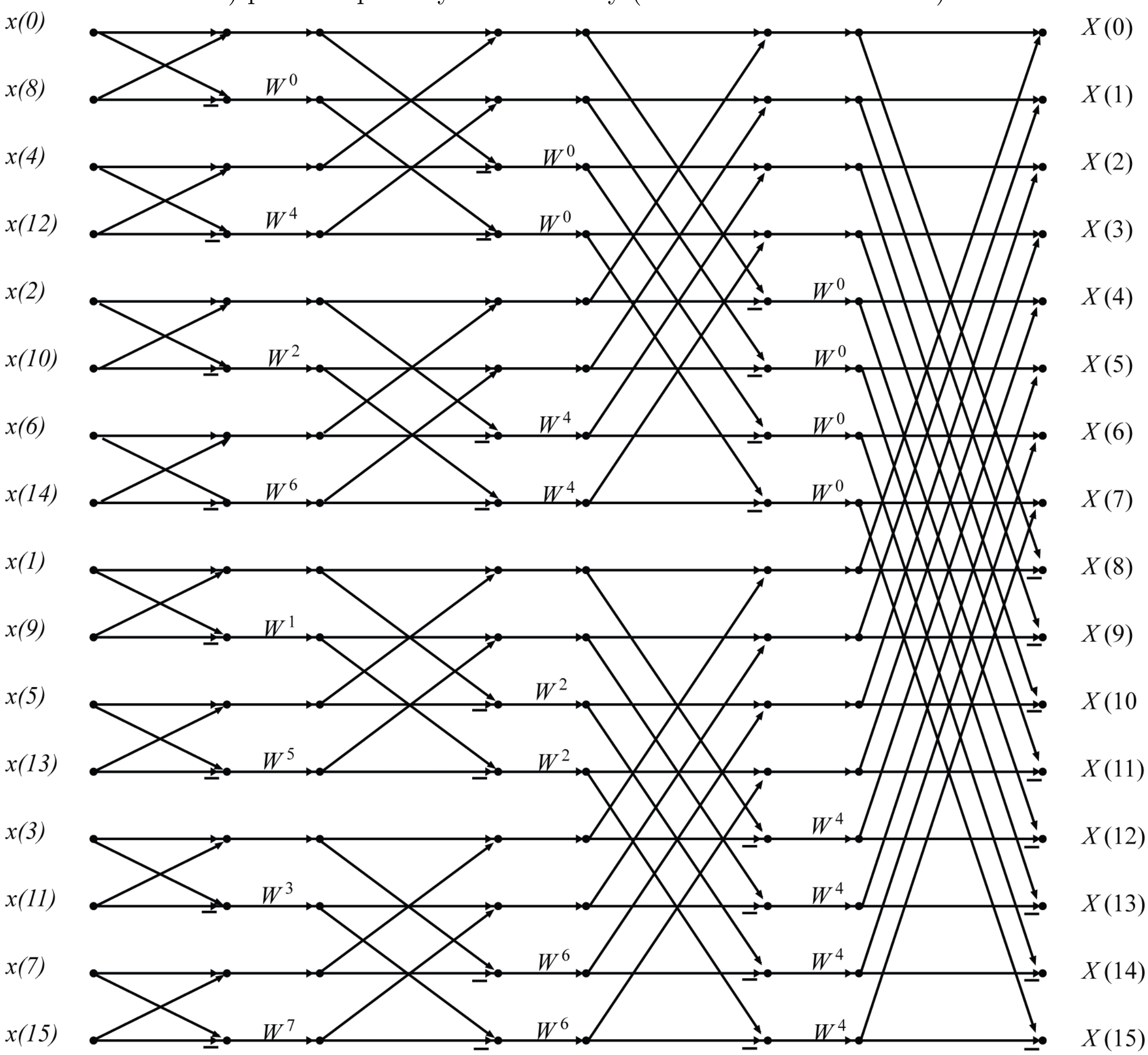
1. Analog circuit of microphone and non inverting amplifier:-
We made a basic circuit for taking audio inputs using a microphone and amplified it by a non inverting amplifier which was made using an op- amp LM324. The basic microphone circuit was made as given below and the output was fed to the amplifier.





Then the output signal was given to the 10-bit ADC inside the FPGA board.
2. Analog Digital converter:-
The FPGA board that we had contained an internal 10-bit ADC and we used the same to convert the analog audio input to digital values to feed inside the fast fourier transform code written in the ISE Design Suite 14.7.
There are 8 analog input-output pins in MOJO.

3. Butterfly Diagram
The algorithm on which the coming values are treated is described below.In the context of fast Fourier transform algorithms, a butterfly is a portion of the computation that combines the results of smaller discrete Fourier transforms (DFTs) into a larger DFT, or vice versa (breaking a larger DFT up into subtransforms).
It recursively breaks down a DFT of composite size n = rm into r smaller transforms of size m where r is the "radix" of the transform. These smaller DFTs are then combined via size-r butterflies, which themselves are DFTs of size r (performed m times on corresponding outputs of the sub-transforms) pre-multiplied by roots of unity (known as twiddle factors).



The benefit of using FFT over DFT is we need to compute only nlogn calculations as compared to n square calculations.The N Log N savings comes from the fact that there are two multiplies per Butterfly.
4. Output on led matrix:-
The various coefficients that we got from the fourier transformation then had to be displayed on a led matrix. We used 8*8 matrixes for the purpose . Then using persistence of vision we displayed the values on the led matrix. Persistence of vision involves lighting up of one LED at a time but

all the LEDs required to be on are lighted one by one very fast so that they look to be lighted simultaneously. Hence the idea of POV was implemented to display the output on an LED matrix. The advantage of using POV is very straight forward. For example, if we had to control an 8*8 matrix, all LEDs individually we needed 64 output pins. But using POV we can do it using only 16 output pins of which 8 would be for the rows and 8 for the columns. And we can light one LED by supplying power to the corresponding row and column only. Then changing it very fast we can achieve any group of on LEDs on a matrix using only 16 pins instead of 64.

6 Limitations and Future Scope

We are able to display output by breaking analog input at 16 different points and then by applying Fast Fourier Transform. It could be optimised by taking analog values at more points like 1024 and hence results could be more accurate.
Morever output could be shown at VGA display which looks more convincing than LED Matrix which we were unable to do due to shortage of time.

7 References

1. *http://embeddedmicro.com/tutorials/mojo/*

   To learn about specifications of MOJO and how to use it and how to install the ISE Design Code and how to code in Verilog.

2. *Advanced Engineering Mathematics by Crezig, Wikipedia*

   To learn about Fourier Transformations from the very basic.

3. *http://en.wikipedia.org/wiki/Sampling*

   To learn about sampling of audio signals.

4. *www.fpga4fun.com/*

   To learn about how to use FPGA board.

5. *Youtube*

   Tutorials on Verilog,FPGA , VGA and LaTex.

6. *www.cosmiac.org/pdfs/tutorial13.pdf, www.lslwww.epfl.ch*

   To learn about VGA from very basics.

8 GitHub link for the Project

https://github.com/harshagarwal1303/Music-visualiser-using-FPGA