Department of (ECE) & (CS), Florida Institute of Technology, Melbourne, Florida

Wake-Up-Word Speech Recognition in FPGA

ECE 5570

1st December 2019

By: Adolf A D'costa

ID: 9035384040

Prof: Dr. Veton Kepuska

Index

1. Project Goal……..3
2. Results Archived……..……....3
3. Pre-requisites to run the code……...3
4. FPGA……..……..4
5. Operational Block Diagram…….……4
6. Description of Code..……...4
7. Pin Planner in Cyclone 4……..……6
8. Resources Used...……7
9. Concept to Implement Project.……8
10. Result……9
11. Conclusion……..……10
12. Reference……10

Project Goal –

The objective of this course was to a system that can detect human voice. The signal is processed on the FPGA to deliver an output. The system is designed to capture a small window speech signal; this signal is processed in several steps to deliver an output.

Results Archived –

Successfully recognized “One” and “Zero” on an FPGA and displayed results on a seven-segment display, the microphone connected to a computer captured speech signal. The entire system is divided into two parts front-end consist of a computer connected to a microphone running a MATLAB code, and the back-end consists of a DE2i-150 FPGA running a code. The front and back end is connected via RS232 Serial port.

Pre-Requisites-

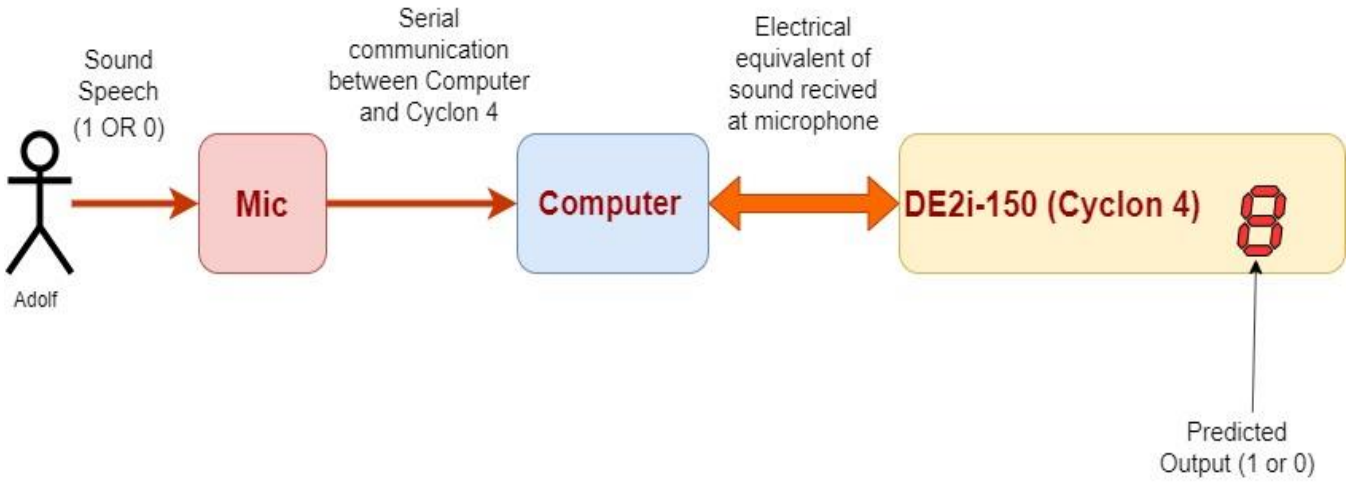
Front-End
MATLAB
“Recorder.m” is the file running in MATLAB

Back-End
DE2i-150 (Development Kit – Hardware)
“Voice_Recognition” is the main file running on the FPGA
UART
UART_parity
UART_Tx
UART_Rx

FPGA–

In this project, we are using the DE2i-150 Development Board. This development board consists of two sections one containing an Intel Atom processor Cedarview (N2600), which is not used. However, on the other side, it has Altera Cyclone 4 (EP4CGX150DF31). The Altera Cyclone 4 is configured to process the signal received via a serial port.

Operational Block Diagram –



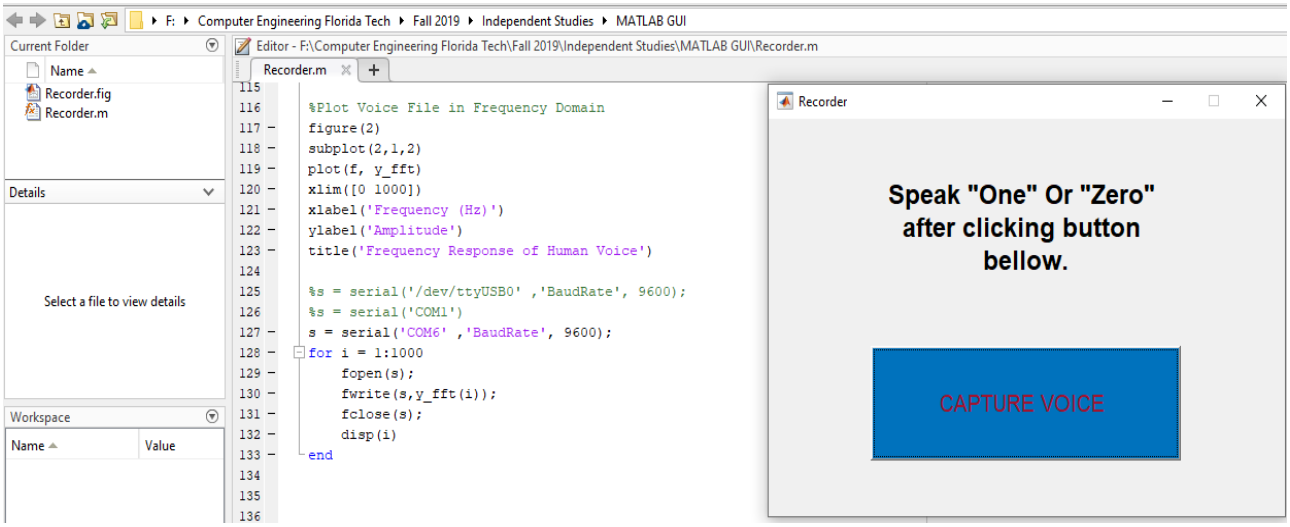
Description of Code –

This code is executed, and the speech signal is analyzed in real-time. The system is divided into two parts front-end and back-end. The front-end has a MATLAB GUI that a “Capture Voice” button, when pressed, picks up a voice sample of 1 second, either “One” or “Zero” from the microphone spoken by the user, which is connected on the computer. The data is stored in a double precession array, Fourier Transform and windowing technique is applied to the signal using MATLAB. Only the first 1000 samples are sent from the entire data for processing and comparison in the FPGA. MATLAB displays a graph of Human voice (Time vs. Amplitude) and Frequency response of the voice signal, i.e.

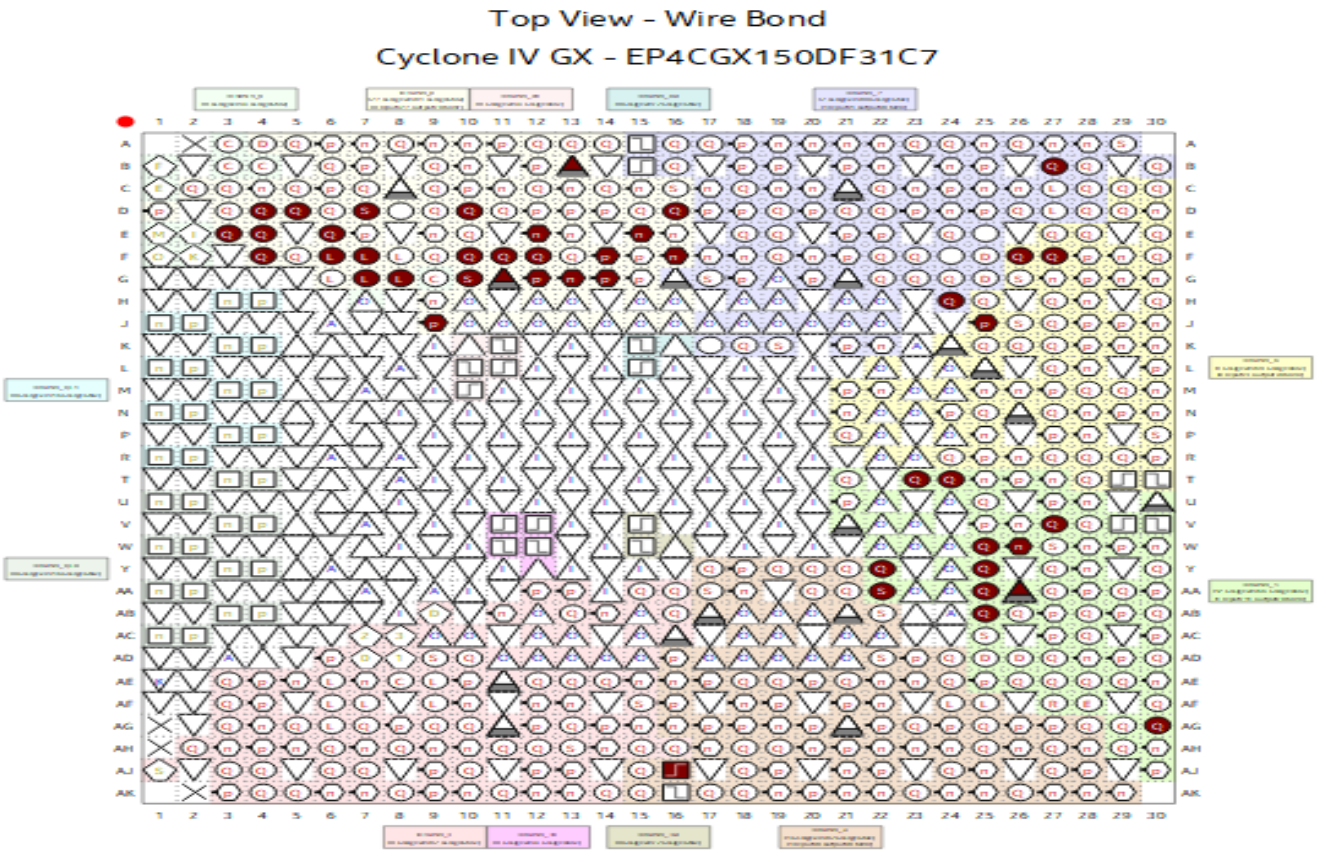
(Frequency vs. Amplitude). USB to Serial converter is used to establish serial communication between the computer and the FPGA. Below is the configuration used for serial communication.

Criteria	Value
Baud Rate	9600
Parity Bit	None
Stop Bit	1
Flow Control	Hardware

The FPGA receives the data is compared with the stored vector. FPGA processes the Euclidian distance, and the weight is calculated. The result is displayed on the seven-segment display. The back-end system runs on four steps Receiving, Calculating distance, Decision making, and Displaying results.



Pin Planner in Cyclone 4 –



Node Name	Direction	Location	VREF Group	Filter location
BUSY	Output	PIN_J25	B6_N0	PIN_J25
CLK	Input	PIN_AJ16	B4_N2	PIN_AJ16
data_led[7]	Output	PIN_AA22	B5_N1	PIN_AA22
data_led[6]	Output	PIN_Y25	B5_N2	PIN_Y25
data_led[5]	Output	PIN_Y22	B5_N1	PIN_Y22
data_led[4]	Output	PIN_W26	B5_N0	PIN_W26
data_led[3]	Output	PIN_F26	B6_N0	PIN_F26
data_led[2]	Output	PIN_F27	B6_N0	PIN_F27
data_led[1]	Output	PIN_AB25	B5_N2	PIN_AB25
data_led[0]	Output	PIN_AA25	B5_N2	PIN_AA25
equal[6]	Output	PIN_G10	B8_N2	PIN_G10
equal[5]	Output	PIN_J9	B8_N2	PIN_J9
equal[4]	Output	PIN_G12	B8_N1	PIN_G12
equal[3]	Output	PIN_F12	B8_N1	PIN_F12
equal[2]	Output	PIN_G13	B8_N0	PIN_G13
equal[1]	Output	PIN_B13	B8_N0	PIN_B13
equal[0]	Output	PIN_G14	B8_N0	PIN_G14
FRAME_ER R	Output			PIN_L30
n[6]	Output	PIN_D4	B8_N1	PIN_D4
n[5]	Output	PIN_D5	B8_N2	PIN_D5

n[4]	Output	PIN_E3	B8_N2	PIN_E3
n[3]	Output	PIN_E4	B8_N2	PIN_E4
n[2]	Output	PIN_E6	B8_N2	PIN_E6
n[1]	Output	PIN_D7	B8_N1	PIN_D7
n[0]	Output	PIN_D10	B8_N2	PIN_D10
o[6]	Output	PIN_F10	B8_N2	PIN_F10
o[5]	Output	PIN_F4	B8_N2	PIN_F4
o[4]	Output	PIN_F6	B8_N2	PIN_F6
o[3]	Output	PIN_AG30	B5_N2	PIN_AG30
o[2]	Output	PIN_F7	B8_N2	PIN_F7
o[1]	Output	PIN_G7	B8_N2	PIN_G7
o[0]	Output	PIN_G8	B8_N2	PIN_G8
result[6]	Output	PIN_F14	B8_N0	PIN_F14
result[5]	Output	PIN_D16	B8_N0	PIN_D16
result[4]	Output	PIN_F16	B8_N0	PIN_F16
result[3]	Output	PIN_F11	B8_N1	PIN_F11
result[2]	Output	PIN_G11	B8_N1	PIN_G11
result[1]	Output	PIN_E12	B8_N1	PIN_E12
result[0]	Output	PIN_E15	B8_N0	PIN_E15
RST_N	Input	PIN_AA26	B5_N2	PIN_AA26
state_four	Output	PIN_W25	B5_N0	PIN_W25
state_one	Output	PIN_T23	B5_N0	PIN_T23
state_three	Output	PIN_V27	B5_N0	PIN_V27
state_two	Output	PIN_T24	B5_N0	PIN_T24
UART_RXD	Input	PIN_B27	B7_N0	PIN_B27
UART_TXD	Output	PIN_H24	B7_N0	PIN_H24

Resources Used –

- RAM consumption approx. 380MB
- Approximately 13,757 Logic Elements Consumed.
- 9144 Registers
- 10,450 Logic Function used
- Accuracy is not static but approx. 60%
- Can detect a speech of either (“One” or “ZERO”) [6]

Concepts Used to Implement Project –

- Discrete Fourier Transform (DFT) – It is important in signal processing. (DFT) is used to converts a finite sequence of equally-spaced samples of a function into an equivalent-length sequence of identically-spaced samples of the discrete-time Fourier transform (DTFT).[1]

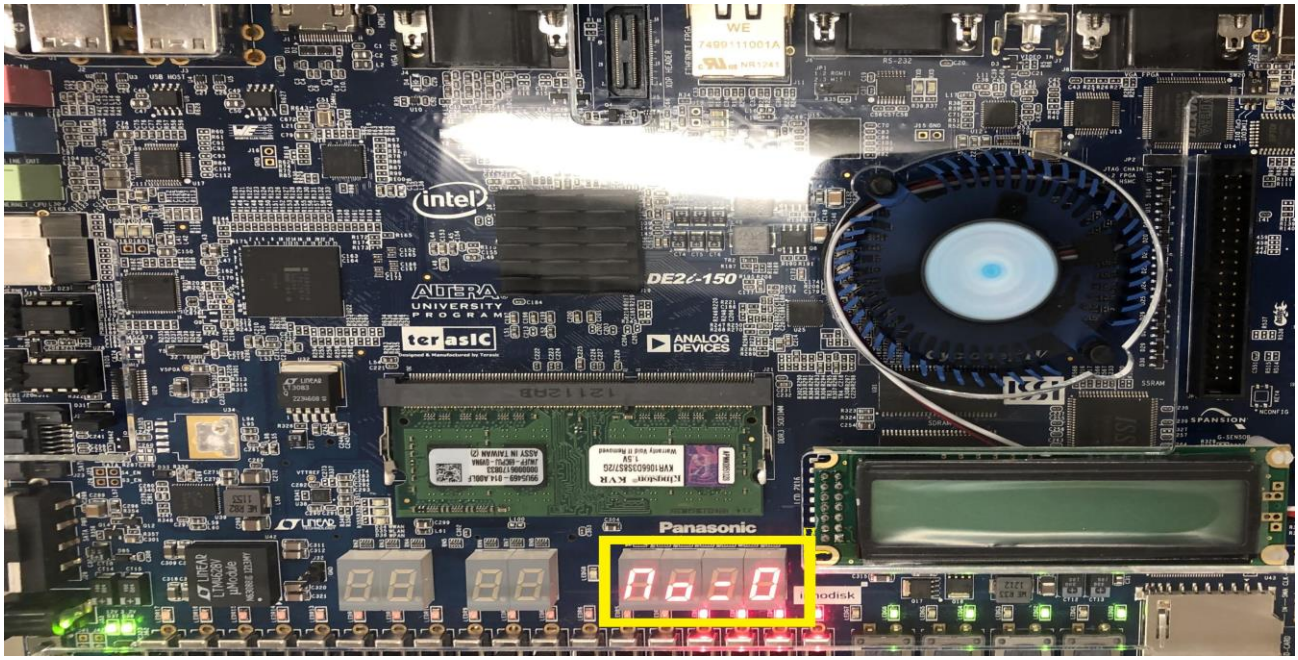
- Hamming Window – Fourier transform applies an infinite number of repeating signals. If the start and the end of the signals did not match, it will generate discontinuity and result in noise. Windowing helps match the edges and keep the waveform smooth, which reduces spectral leakage.

- Euclidian Distance – It is an ordinary straight line distance between two points in Euclidian Space.

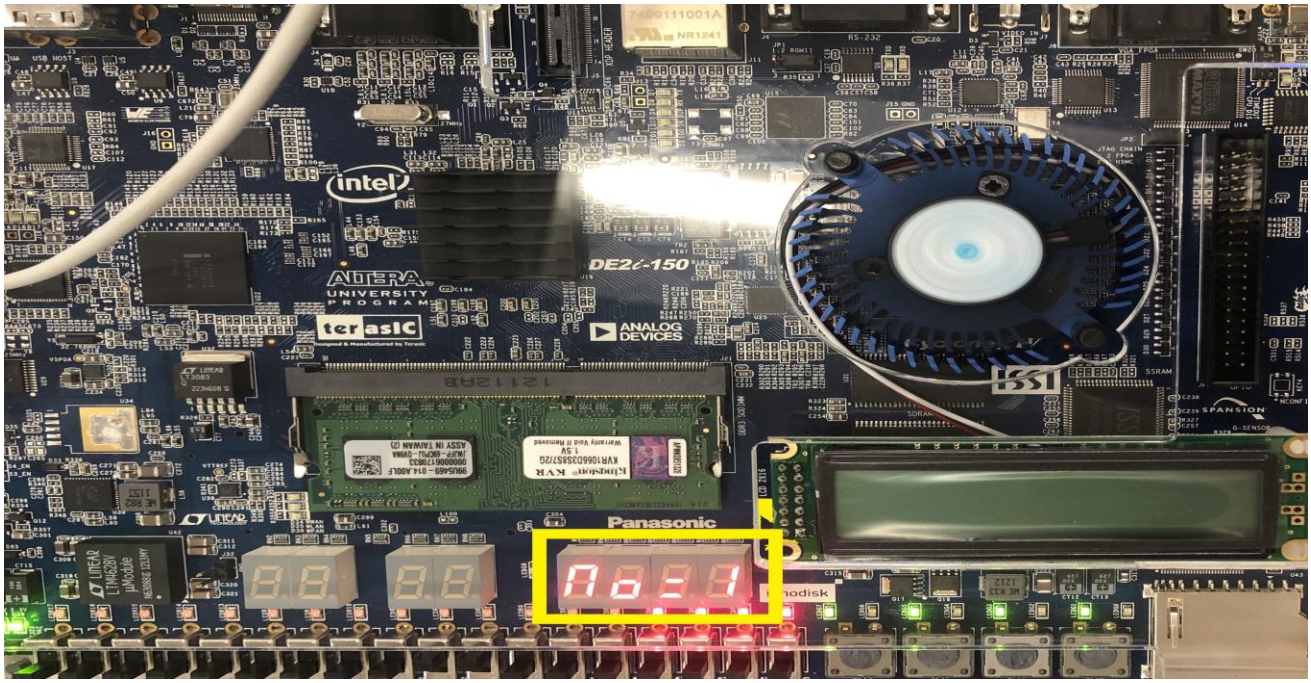
- Hamming Distance – It is the distance between two strings of equal length. It is the minimum number of substitutions required to change one string into another.

- FFT- It is a O[N log(N)] is used to compute Discrete Fourier Transform. It converts a signal from (time or space domain) to the frequency domain. [2]

Result –



"Zero" was detected by the FPGA



"One" was detected by the FPGA

Conclusion –

Successfully was able to detect speech signals using the Altera DE2i-150. This FPGA has many constraints and incapability. The software was used to overcome this hardware incapability; that's the reason why the system had to be divided into two front-end and the back-end. The system was successful in recognizing speech signal i.e., "One" and "Zero," the efficiency of the system was not as good as a processor-based system. The availability of powerful hardware with powerful ADC chips can record more pure sound resulting in more accurate results; powerful chips can run robust algorithms like Hidden Markov models.

Reference –

[1] "Discrete Fourier Transform." Wikipedia, Wikimedia Foundation, 6 Dec. 2019, https://en.wikipedia.org/wiki/Discrete_Fourier_transform.

[2] "Fast Fourier Transform." Wikipedia, Wikimedia Foundation, 16 Nov. 2019, https://en.wikipedia.org/wiki/Fast_Fourier_transform.

[3] "Field-Programmable Gate Array." Wikipedia, Wikimedia Foundation, 30 Nov. 2019, https://en.wikipedia.org/wiki/Field-programmable_gate_array.

[4] "Choosing a Wake Word - Tips & Considerations." Picovoice, https://picovoice.ai/docs/choose-wake-word/index.html.

[5] Kepuska, Veton. "Wake-Up-Word Speech Recognition." IntechOpen, IntechOpen, 13 June 2011, https://www.intechopen.com/books/speech-technologies/wake-up-word-speech-recognition.

[6] "Altera DE2 Board - Mạch Thí Nghiệm FPGA." Altera DE2 Board - Mạch Thí Nghiệm FPGA, http://kitboardmach.blogspot.com/2017/11/altera-de2-board-mach-thi-nghiem-fpga.html.

[7] DE2i-150 FPGA Development Kit, 17 May 2019, https://www.intel.com/content/www/us/en/programmable/solutions/partners/partner-profile/terasic-inc-/board/de2i-150-fpga-development-kit.html.

[8] MohammedRashad. "MohammedRashad/FPGA-Speech-Recognition." GitHub, https://github.com/MohammedRashad/FPGA-Speech-Recognition.