REAL TIME SPEAKER RECOGNITION on TMS320C6713

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

FOR THE REQUIREMENTS OF

BACHELORS IN TECHNOLOGY

BY

ABHIJIT TRIPATHY 108EC013

&

AKASH SAHOO 108EI010

UNDER THE GUIDANCE OF

PROF. A.K Swain

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

2008-2012

NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA

CERTIFICATE

This is tocertify that the thesis titled , "***Real Time SpeakerRecognition on TMS320C6713"*** submitted by **MR.ABHIJIT TRIPATHY (108EC013)** and**MR.AKASH SAHOO (108EI010)** in partial fulfillments of the requirements of **Bachelors in Technology** degree in **ELECTRONICS AND COMMUNICATION**and **ELECTRONICS AND INSTRUMENTATION** , National Institute ofTechnology,Rourkela is an authentic work carried out by them under mysupervision.

DATE: **PROF. Ayas Kanta Swain**

Department of Electronics and Communication

CONTENTS

ACKNOWLEDGEMENT

We would like to take this wonderful opportunity of doing BTech Project on **"Real Time Speaker Recognition on TMS320C6713"** which has given us a lot of learning and practical exposure.

First of all,we would like to express our gratitude to our guide,**prof.T.K. Dan** who gave us the opportunity to work on this project.He gave us independence indoing whateverwe liked .He really inspired us throughout the project.Secondly, we would like to thank**Prof A.K. Swain**who has been the guiding force in our project.His constant motivation has been of great help to us.

We would also like to thank **Prof. U.K. Sahoo**who has helped us with signalprocessingconcepts on which we needed clarification.

Lastly we would like to thank**Mr.Sumit Kumar Sao**,**Mr. Kanhu Charan Bhuyan,Mr .Venkat ,Mr Karupannan** who have provided us with all the things that we needed in **Embedded Systems Lab.**

ABHIJIT TRIPATHY **AKASHSAHOO**

108EC013 **108EI010**

ABSTRACT

Speaker recognition is defined as the process of identifying a person on the basis of the information contained in speech.In this world where breach of security is a major threat ,speaker recognition is one of the major biometric recognition techniques. A large number of organizations like banks,defence laboratories , industries , forensic surveillance are using this technology for security purposes.

Speaker recognition is mainly divided into two categories **:** Speaker identification andSpeaker verification. In speaker identification we find out which speaker has uttered the given speech ,whereas in speaker verification we determine if the speaker who is claiming a particular identity is telling the truth or not.

In our first phase we did speaker recognition on MATLAB.The process we followed comprised of three parts.First we did preprocessing where we truncated the signal and performed thresholding on it.Then we extracted the features of speech signals using Mel frequency Cepstrum coefficients. These extracted features were then matched with a set of speakers using a Vector Quantization approach.

In our second phase we tried to implement speaker recognition in real time.As speaker recognition is a signal processing task ,we decided to implement it real time on a DSP (digital signl processor) as it performs very fast multiply and accumulate operations(MAC) and speaker recognition had stages where signals were primarily added and multiplied .Hence DSP was choosen as our platform.The second phase comprises our familiarisation with the TMS320C6713 DSP,the first few audio applications we performed on it,some real time filters we developed on it and finally our speech recognition problem.

MOTIVATION

Suppose we are having audio data which is recorded every day for several years. Out of these data we want to find out which one corresponds to a particular speaker.This can be done by using a speaker recognition system. Consider a case of video conferencing, where we want to focus a camera on a particular person . This can be easily implemented if a speaker recognition system exists.

Automatic speaker verification and automatic speaker identification systems are the most economical solution to the problem of unauthorized computer access and other digital equipments hacking involving communications. As we know that almost all computers have microphone and telephone networks, so.speaker recognition can be implemented in software .This reduces hardware complexities .Moreover speaker recognition systems can be made pretty robust to noise, mimicry and environmental changes.

LITERATURE SURVEY

A lot of work on speaker recognition has already been done in industry, technological labs and educational universities. For example **AT&T** labs have synthesized speaker recognition systems.The **NIPPON Telephone and Telegraph Company** have their own speaker recognition systems. **MIT** of USA and **National Tsing Hua university** of Taiwan, **Texas Instruments** also have conducted testing of various speaker recognition systems. Automatic speaker identification systems applications include access control, telephone banking and telephone credit cards.**ITT, Lernout & Hauspie,T-Netix** etc. are known for their automatic speaker verification systems.

It is estimated that cyber crimes involving 3-4 million dollars occur in banks in USA every year.Imagine how much can we save by using speaker recognition systems to check these fraudulent transactions.Products like **SPRINTS's Foncard**which uses **TI's**speaker verification engine are used to check the occurrence of these kinds of scams.

CHAPTER 1

INTRODUCTION TO SPEAKER RECONITION

Speech is an arrangement of notes that will never be played again.

F. Scott Fitzgerald

1.1 Introduction

Speaker recognition is the process of automatically recognizing who is speaking on the basis of the information contained in the speech signal. Speaker recognition is carried out in two phases.First phase comprises the training phase whereas, the second phase comprises thetesting phase. In the training phase the speaker has to utter something so as to provide speech samples.These samples enable the system to build a model of that speaker.In the testing phase ,the input speech is matched with the stored reference models and a decision is made. Speaker recognition can be said as a biometric system which validates a person's claim to an identity based on the features extracted from the speech samples. The training and testing phase of speaker recognition system makes it possible to use the speaker's voice to verify his/her identity and control access to services such as voice dialing,telephone banking,telephone shopping,voice mail and other computer access services that are voice activated.

1.2 Principles of Speaker Recognition

Speaker recognition can be classified into two categories: identification and verification.

Speaker identification is the process of determining which registered speaker provides the utterance.

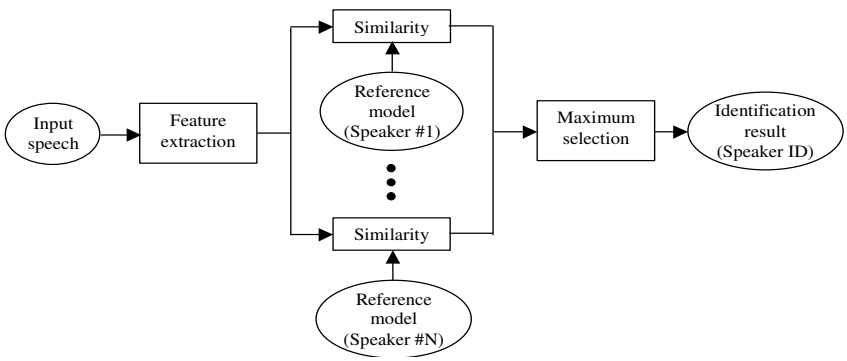


Figure 1. depicting SI

Speaker verification on the other hand is a process of determining whether the claim of the speaker is true or not.

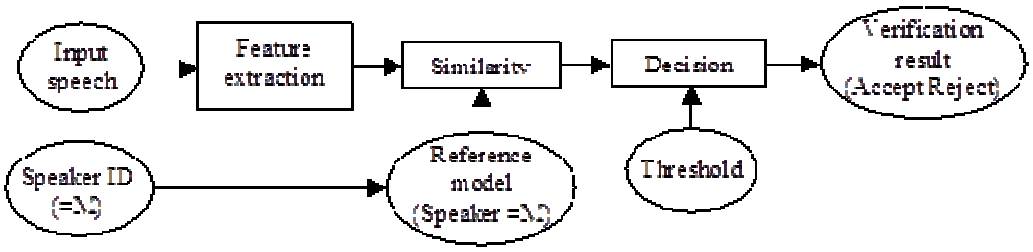


Figure 2 depicting SV

However we find out that speaker recognition is a difficult task. Speaker recognition assumes that human voice exhibits characteristics that are unique to the speaker. This principle is challenged by the highly variant nature of speech signal. Speech signals in training and testing can vary greatly due to time dependent changing of voice , health conditions ,acoustic noise variations in the environment etc. .

1.3 Speech's Feature Extraction

The feature extraction process converts the speech waveform into a parametric representation for analysis.Feature extraction is also named as signal processing front end.

The speech signal is a slowly time varying signal.Speech characteristics are fairly stationary over short period's of time i.e. between 5 to 100 ms.However over longer periods of time i.e. over 200ms ,the characteristics vary due to the different speech spoken. Therefore STFT(short time fourier transform) is used.

A wide range of possibilities exist representing the speech signal for the speaker recognition task, such as Linear Prediction Coding (LPC), Mel-

Frequency Cepstrum Coefficients (MFCC), and others. MFCC is perhaps the best known and most popular, and will be described in the following sections.

MFCC's are based on the known variation of the human ear's critical bandwidths with frequency with filters spaced linearly at low frequencies and logarithmically at high frequencies being used to capture the phonetic characteristics of speech. This is expressed in the mel-frequency scale, which is a linear frequency spacing below 1kHz and a logarithmic spacing above 1 kHz. The process of computing MFCCs is described in more detail in MFCC processor section .

The process of feature extraction is divided into 2 parts -- **Preprocessing & Feature Extraction.**

The task of preprocessing consists of the following stages :

- Truncation
- Frame Blocking
- Windowing
- Short time Fourier Transform

1.4 Preprocessing

1.4.1 Truncation

In MATLAB , the default sampling frequency of wavread command is 44100 Hz.Now if we record an audio clip using wavread for say,3 seconds we would have 132300 samples. Many of these are unnecessary data values as the speaker may not be speaking for some time ,which may have been sampled. In order to remove these unnecessary samples , we truncate the recorded data by applying a particular threshold.

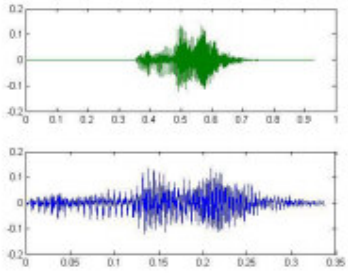


Figure 3 depicting truncation

1.4.2 Frame Blocking

In frame blocking step,we block the input ,continuous speech sample into a finite number of frames.Consider we block the speech into frames with M data samples with adjacent frames separated by N sampled datas.Here,the first frame consists of N data elements and the second frame begins with the Nth data sample of the first frame .The second frame overlaps the first frame by N-M samples and the third frame overlaps the second frame by N-M samples and so on.

1.4.3 Windowing

After frame blocking we go for windowing. In windowing we multiply each frame by the window function so as to minimize the signal discontinuities at the beginning and end of each frame. The idea is to set the signal value to zero in the beginning and end of each frame.Window function w(n) is defined for $0 \leq n \leq N-1$,where N denotes the number of elements in each frame.

We used a Hamming window for minimizing the interference from side lobes.The Hamming window has the formula $w(n) = 0.54 - \cos\left(\frac{2\Pi n}{N-1}\right)$

,for $0 \leq n \leq N-1$.

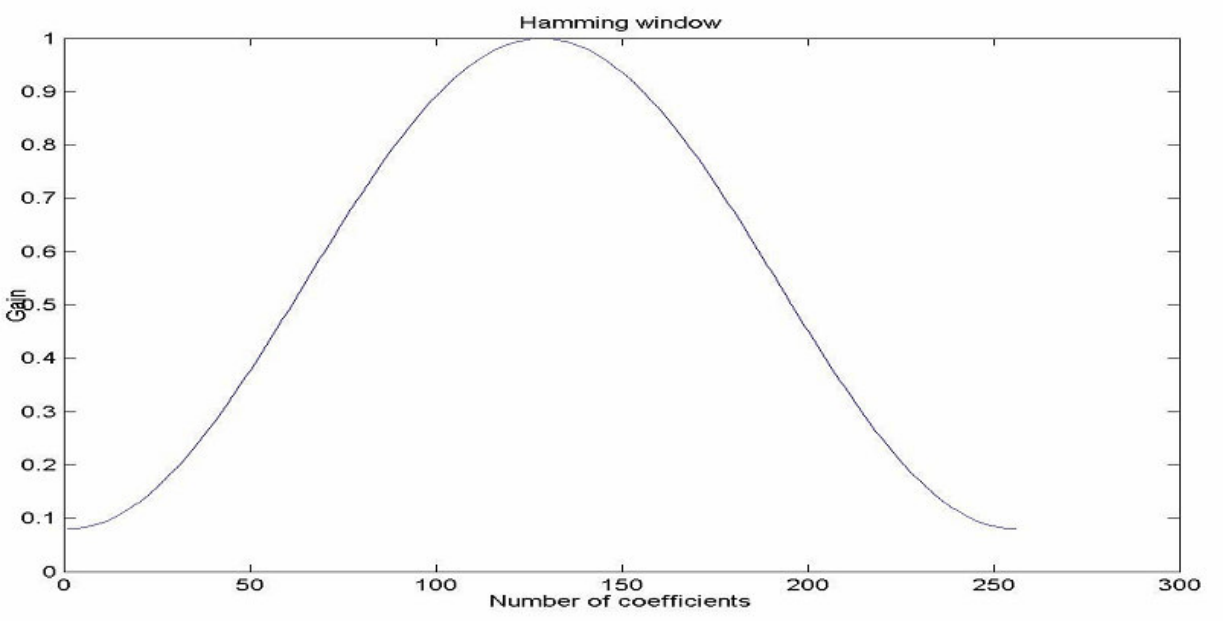


FIGURE 4 depicting a Hamming Window

1.4.4 STFT – Short Time Fourier Transform

Then we compute the fast Fourier transform which maps each frame onto the frequency domain.The FFT actually reduces the number of calculations required to compute the DFT of a given set of samples.For a set of N samples the STFT is defined as

$X(K) = \sum_0^{N-1} x(n)e^{-j2\pi kn/N}$,where $0 \leq K \leq N-1$

Here the X(K) values are complex numbers and we consider only their absolute values.The sequence of X(K) defined as$\{X_k\}$ is related to frequency as per the following formula.

Positive frequencies $0 \leq F \leq Fs/2$ correspond to $0 \leq n \leq N/2-1$

Negative frequencies $-Fs/2 \leq F < 0$ correspond to $N/2 \leq n \leq N-1$

Where Fs is the sampling frequency.

This FFT calculation is also called as spectrum or periodogram.

1.5 Feature Extraction

1.5.1 Cepstral Coefficients UsingDiscrete Cosine Transform

In preprocessing we reduced the number of samples of speech by truncation .By frame blocking we further reduced the amount of data we handle at a time by dividing the speech signal into frames. To extract the **Cepstral** features we employ the following formula

$$ceps=DCT(\log(abs(FFT(y_{windowed}))))$$

Only the first 10 cepstralcoefficents are considered for the process of feature matching.

1.5.2 Mel Frequency Wrapping

Psychological studies have found out that our perception of frequency contents for speech signals doesn't follow a linear scale. Due to this,for each tone with actual frequency f(in Hz), a subjective pitch is found out and is measured on a scale known as **mel scale.** The mel scale is a special scale.It has a linear frequency spacing for frequencies upto 1 kHz and above that it has a log scale.

To create a virtual subjective spectrum we use a mel filter bank.For computational simplicity we use a triangular band pass filter whose bandwidth is determined by a mel frequency interval which is typically a constant. We chose the number of melcepstrum coefficients to be say,20.We have shown below a triangular mel band pass filter.We can visualize this filter as a histogram bin in frequency domain.The mel scale is represented as $f = \frac{1000}{log2} \times \log\left(1 + \frac{f}{1000}\right)$

.

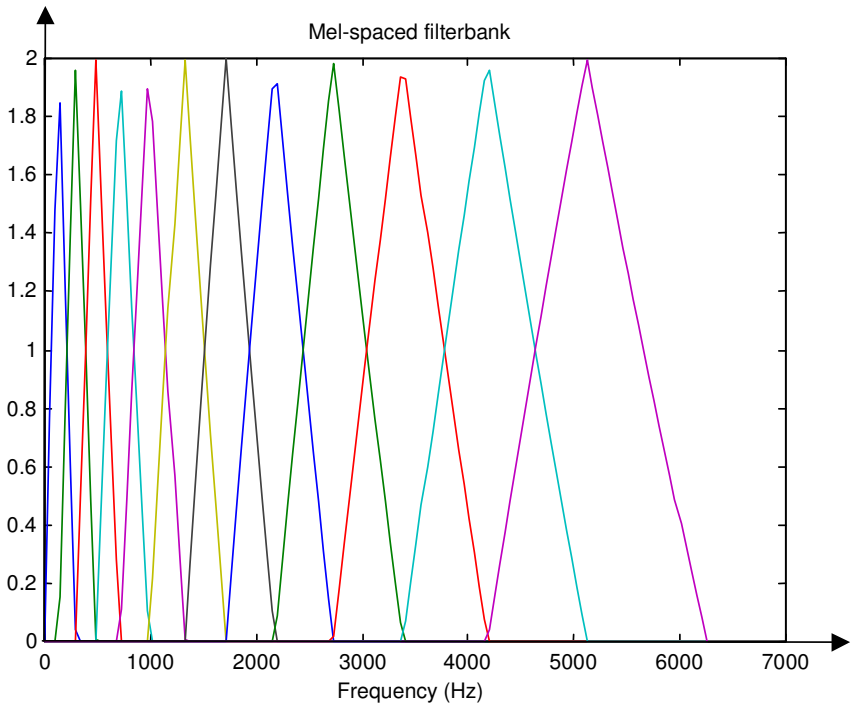


Figure5showing a mel filter bank for 20 mel coefficients

1.5.3 MFCC Cepstral Coefficients

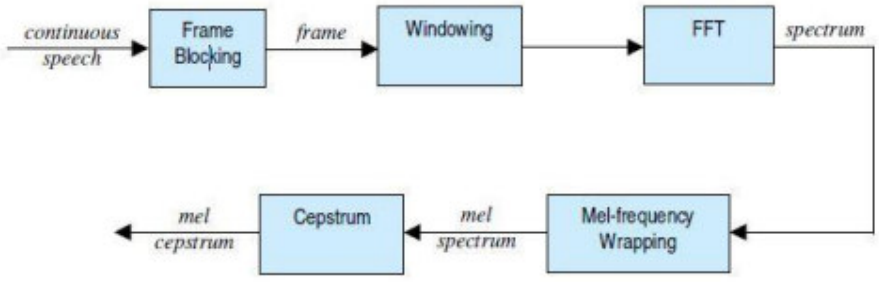


Figure 6 – MFCC Steps

The above figure shows the block for computing the MFCC (mel frequency cepstrum coefficients).If convert the log mel spectrum back to time the result we get are called MFCCs.Thecepstral representation of speech gives a good

representation of the local spectral properties of the signal for a given frame analysis.

The mel spectrum coefficients being real quantities are convertible to time domain using **discrete cosine transform**.

If the absolute of the fft coefficients are given as S_k where $0 \leq k \leq K-1$ then the mel cepstrum coefficients are given as C_n ,where

$$C_n = \sum_{k=1}^{K} \log(S_k) \cos\left\{ n\left(k - \frac{1}{2}\right)\frac{\pi}{k} \right\},\ n = 0,1,\ldots K-1$$

We exclude the first value since it represents the mean of the signal and is of little importance.

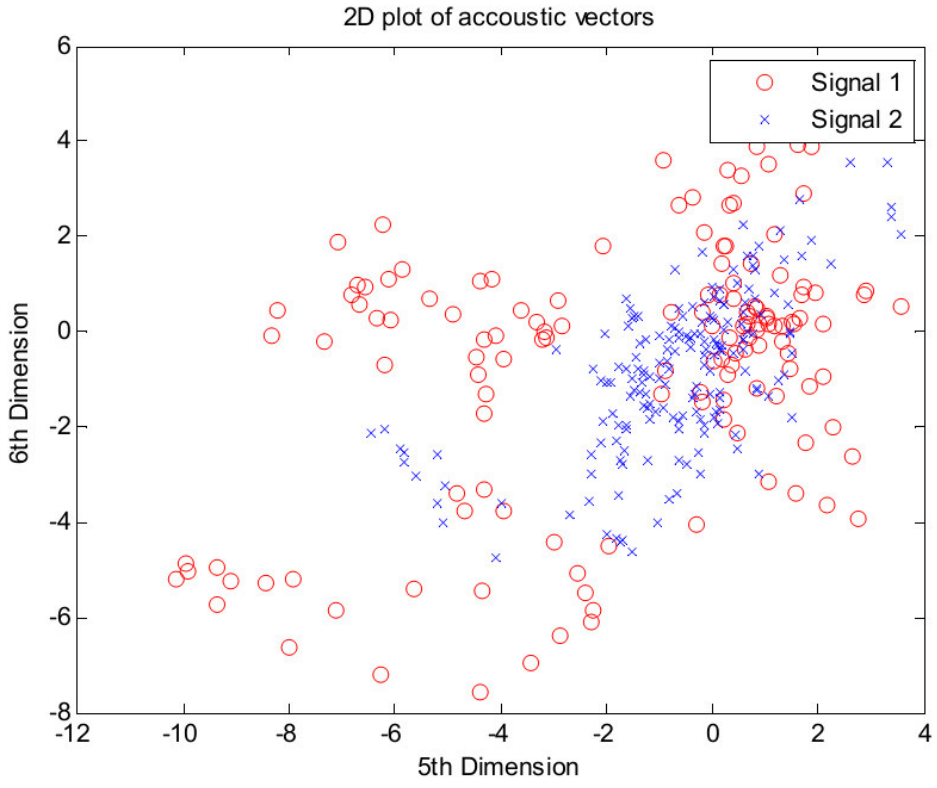


Figure 7 showing acoustic vectors corresponding to 5th and 6th filters of mel filter bank.

1.5.4 Main points of feature extraction :

After following the above instructions , we obtain a set of mel-frequency cepstral coefficients. It takes into account a frame overlap of about 30 ms. These set of mel-frequency coefficients are named as acoustic vectors. Similarly we find the acoustic vectors for each speech signals.

1.6 Feature Matching

Speaker recognition problem comes under the category of pattern recognition. The main aim of pattern recognition is to divide a number of objects into classes.The objects to be classified are called patterns and the process of classifying them is called pattern classification.

In the case of speaker recognition the set of acoustic vectors are the 'patterns'.The individual speakers are our 'classes'. The problem of speaker recognition is our 'pattern classification' problem.

In our case of speaker recognition we follow a supervised learning method.We have a training set of patterns and a testing set. The training set of patterns are used to determine the algorithm to be used for pattern classification. The training set are compared with the testing set for classification.If we know beforehand the correct set of patterns of the testing set ,then we can calculate the accuracy of our algorithm.

The most well known algorithms for pattern recognition used in speaker recognition algorithms are Dynamic Time Warping(DTW),Hidden Markov Modeling(HMM), and Vector Quantization(VQ) .In our project ,the VQ approach is used due to its ease of implementation and lesser complexity and good accuracy.

Vector Quantization is a method of mapping vectors from a large vector space to a definite number of regions in the vector space. Each such region is a cluster .The center of each cluster is called codeword.The collection of all such code- words is called code book.

The figure below shows a conceptual diagram depicting the recognition process. In the figure two speakers and two dimensions of the acoustic space are shown. The circles refer to the acoustic vectors from the speaker 1 whereas the triangles are from speaker 2.In the training phase , using the clustering algorithm described above , vector quantization codebook specific to a speaker is generated. The resulting codewords ,centroids are shown by black circles for speaker 1 and black triangles for speaker 2. The speaker corresponding to the smallest total distortion is known as the utterance's original speaker.

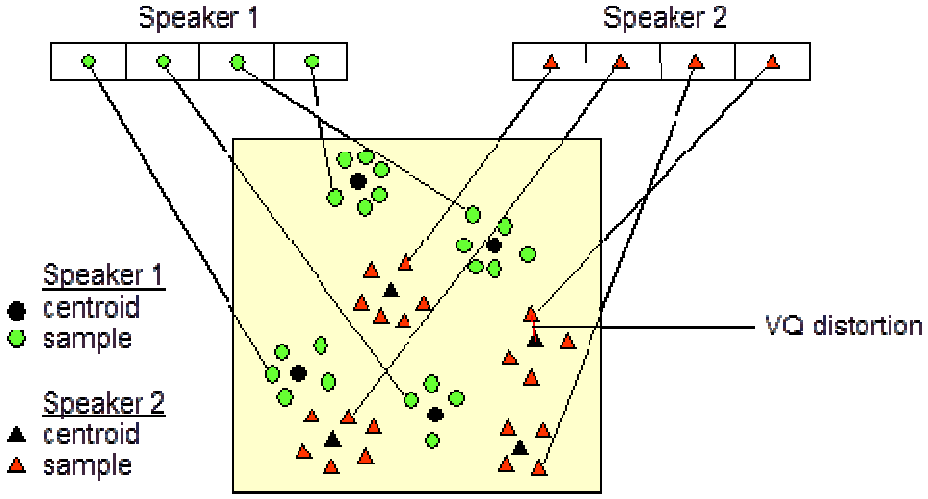


Figure 7 depicting recognition process

The algorithm used to cluster the training is LindeBuzo Gray (LBG) algorithm. Let's illustrate the algorithm to cluster a set of L training and M codebook vectors.

1. Firstly, generate a single vector code book .this is the centroid of the complete set of training vectors.

2. Now having completed the first step, double the size of the code book by separating the present codebooks in accordance with the rule given below.

$$\mathbf{y}_n^+ = \mathbf{y}_n(1+\varepsilon)$$

$$\mathbf{y}_n^- = \mathbf{y}_n(1-\varepsilon)$$

Where,

$1 \leq n \leq$ size of code book

3. Conduct a search in the present code book for the nearest neighbour i.e. the codebook with the most matching characteristics .Then allot the corresponding cell with that vector.
4. Now change the codeword in each cell using the centroid of the training vectors assigned to the particular cell.
5. Repeat the procedures given in steps 3 and 4 unless the mean distance falls below a predefined threshold
6. Repeatedly perform the instructions in steps 2,3 and 4 until a code book of size M is designed

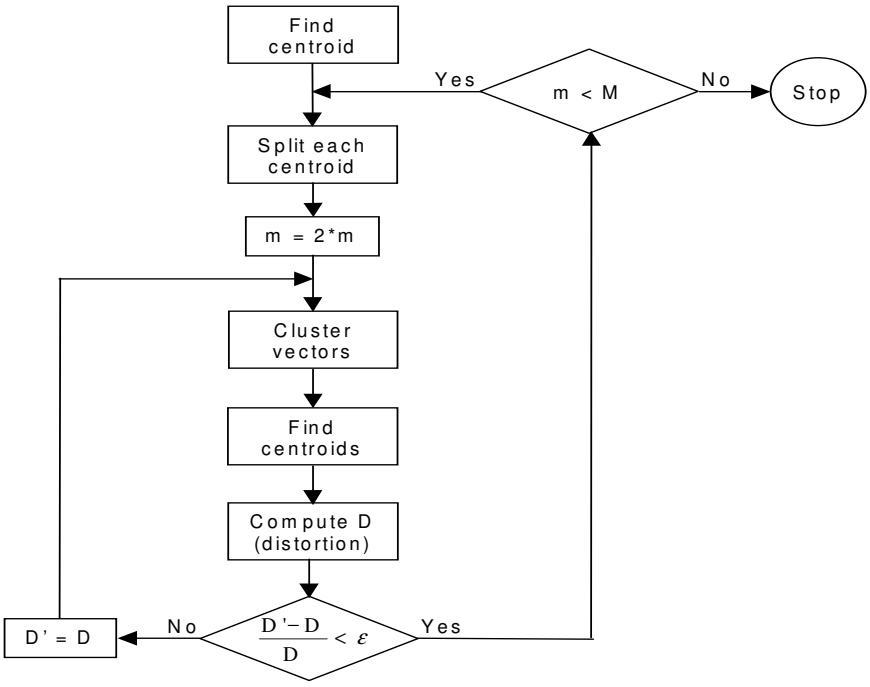


Figure 8 depicting LBG algorithm

1.7 Gaussian Mixture Modeling

We see that when the feature vectors after clustering are examined in a d-dimensional feature they approximate to a Gaussian distribution. This ensures that we can view feature vectors in terms of their probability distributions. We make this assumption due to the following statements :

1. Gaussian classes can be considered to represent set of acoustic classes. They represent vocal tract information.
2. Gaussian mixture density closely resembles the feature vector distribution in a multi-dimensional feature space.

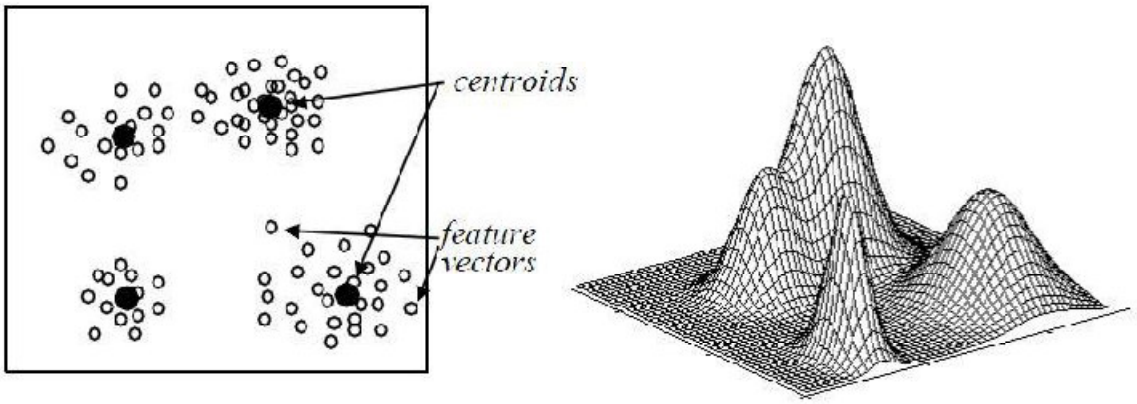


Figure showing Gaussian model of a feature vectors

1.8 Model Description

A Gaussian mixture density can be represented as a sum of M probability densities by the following equation :

$$p\left(\overrightarrow{x}\middle|\lambda\right) = p_i b_i(\vec{x})$$

where$\vec{x}$ refers to a feature vector, p_i represents the mixture component and $b_i(\vec{x})$ is a D- variate equation.

The complete Gaussian mixture density is represented by mixture weights ,mean and covariance of corresponding and denoted as $\lambda = \{p_i, \mu_i, \sum i\} i = 1, \ldots M$.

1.9 Maximum Likelihood Estimations

The expectation maximization algorithm is followed to classify the acoustic vectors. It is defined as :

- Assume initial values of p_i , μ_i,$\sum i$
- Calculate the next values of mean, co- variance and mixture weights iteratively such that probability of classification is maximized.

After completing the modeling of each user's Gaussian mixture density we get a Gaussian distribution representing all components. For k speakers it is given as $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_k\}$ .The recognized speaker, is the one with the maximum value of posterior probability for a given utterance.

CHAPTER 2

DSP Platform & Architecture

Slow & Steady does not wins the race – speed is good

2.1 Introduction

2.1.1 Embedded Computing And Dsps

An embedded system is a computer system designed for a specific function within a larger system, often with real time computing constraints with the major computing unit being microprocessor. In the present day scenario microprocessors are found in

1. Computer system applications-General purpose processors
2. Embedded system applications-Microcontrollers
3. Digital signal processors-DSPs

2.1.2 DSPs

Digital signal processing applications are often viewed as a third category of microprocessor applications because they use specialized CPUs called DSPs and so DSPs are used mainly for signal processing applications. In reality the boundaries of application areas are not as well defined as it may seem . e.g-DSPs may be used for high speed computation such as specialized co-processor boards designed for intensive scientific computation.

Some features of DSP's are :

- Most DSP processors share common basic features designed to support high performance ,repetitive, numerically intensive tasks. The most often cited of these features is MAC operation. The MAC operation is useful in DSP applications where dot product of vectors are involved e.g-digital filters ,FFT etc. .
- The second important feature of DSP is to complete several instructions in a single cycle. This allows the user to simultaneously fetch an instruction while decoding or executing another.
- A third most important feature is the facility of speeding arithmetic processing by having more than one addressing units. Once the appropriate addressing registers have been configured ,the address

generation units form the addresses required for operand access in parallel with arithmetic instruction's execution.

- To allow a low cost high performance input and output ,most DSP processors incorporate one or more serial and/or parallel interfaces and specialized i/o handling mechanisms like DMA and interrupts to proceed without any intervention from the processor.

2.2 Introduction to TI C6000 Architecture

Any processor can be classified by 3 ways

- **Memory-Havard and Von Neuman**

  In Havard program and data memories have different buses, whereas in

  Von Neuman program and data memories share buses.

- Ins**truction –RISC,CISC**

  RISC-reduced instruction set computing

  CISC-complex instruction set computing.

- **Instruction execution-Very long instruction word (VLIW)** enables several instructions running in a single cycle.

2.2.1 Features of TI C6000

- Advances VLIW CPU with eight functional units,including two multipliers and six arithmetic units. Executes up to eight instructions per cycle for up to ten times the performance of DSPs.
- *Instructions packing* Gives code size equivalence of eight instructions executed in serial or parallel. Reduces code size, program fetches and power consumption.
- *All instructions execute conditionally.* Reduces costly branching and increases parallelism.

- Code executes as programed on independent units.
- 8/16/32 bit data support, providing efficient memory support for a variety of applications.
- Saturation and normalization provide support for key arithmetic operations.
- Peak 1336 MIPS at 167 MHz, Peak 1G FLOPS at 167 MHz for single precision operations and Peak 250 MHz for double ones.

2.2.2 CPU

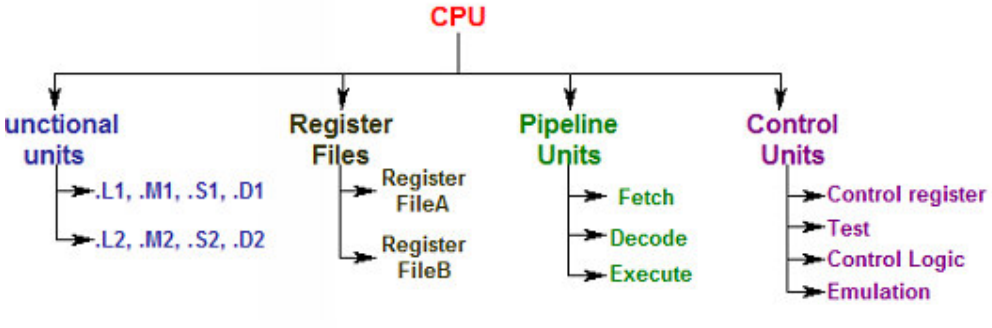


Figure 8 showing a CPU registers & units.

CPU can be divided into four functional parts:

- **Functional units**These are four floating point,two fixed point and two multipliers.
- **Register files** There are two register file each of 16 CPU registers and each register of 32 bit.
- **Pipeline units** In this pail there are 3 units Fetch unit,dispatch unit and decode unit.
- **Control units** It consists of control registers,test,emulator,interrupt control like AMR,CSR,PCE1

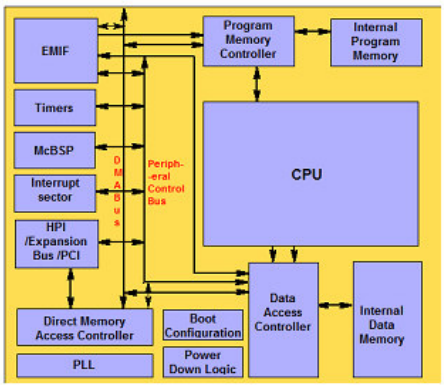
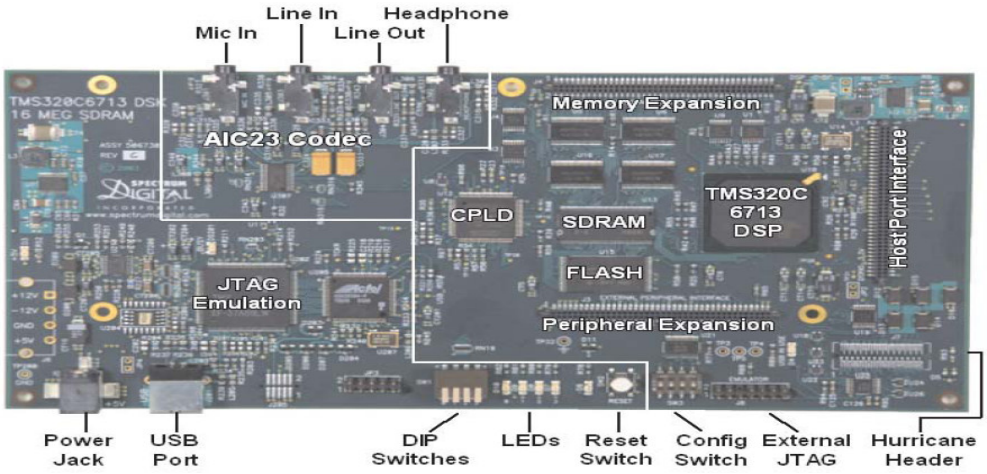


Figure 9 showing components of TI C6000

2.3 TMS320C6713 DSK

2.3.1 introduction



The C6713 DSK is a low cost platform which lets us develop applications for the Texas Instruments C67XX DSP family. The primary features of the DSK are

- 225 MHz TMS320C6713 Floating point DSP
- AIC23 Stereo Codec
- Four Position user DIO switch and four user LEDs
- Onboard Flash and SDRAM

The Code composer studio tools are bundled with the C6713 DSK providing the user with an integrated development environment for C and assembly programing. Code Composer studio communicates with the DSP using an on board JTAG emulator through an USB interface.

The TMS320C6713 is the heart of the system .It is a core member of Texas Instruments C64XX line of fixed point DSPs whose distinguishing features are an extremely high performance 225MHz VLIW DSP core and 256 Kbyes of internal memory. On chip peripherals include a 32-bit external memory interface(EMIF) with integrated SDRAM controller, McBSPs . The 6713 represents the high end of Tis c6700 floating point DSP line both in terms of computational performance and on-chip resources.

The C6713 has a significant amount of internal memory so that many applications have all code and data on-chip. External accesses are done through EMIF.

DSPs are frequently used in audio processing applications so the DSK includes an on board CODEC called e AIC23.Codec stands for coder/decoder. The DSK has 4 light emitting diodes and 4 DIP switches that allow users to interact with programs through simple LED displays and switch inputs.

2.3.2 Detailed Features

The DSK c6713 supports a TMS320C6713 DSP which can operate at a clock frequency of up to 225MHz. The DSP core is designed for high performance floating point operation .Beyond the DSP core, the 6713 integrates a number of on chip resources that improve functionality and minimize hardware developement complexity . Some features are

- VLIW core VLIW is a processor architecture that allows many instructions to be issued(8 in 6713) in a single clock while still allowing for ery high clock rates. VLIW architecture can achieve high processing rate but puts more burden on the compiler to schedule concurrently executed instructions. The code generation tools in CCS are optimized for this.

- 192Kbytes Internal memory for high speed performance
- 64Kbytes L2 cache /RAM four 16K byte blocks of internal RAM that can be configured as RAM or cache.
- 4Kb program /data cache separates caches for program code and data.
- On chip PLL generates processor clock rate from slower external clock reference
- 2 Timers generate periodic timer events as a function of processor clock. They are used to create time slices by DSP-BIOS for multitasking.
- EDMA Controller allows high speed data transfers without intervention from DSP
- 2 McBSPs Multichannel buffer serial ports can be used for high speed dats transmission with external devices or can .be reprogrammed as general purpose i/o .
- 2 McASPs Multichannel audio serial ports are used for audio applications .Not used on the DSK, they are brought out to he expnsiion connectors.
- 2 I2c interfaces Inter integrated circuit bus is a serial format that can support several devices per bus.

EMIF External Memory interface is a 32 bit bus on which external memories and other devices can be connected. It includes features like internal wait state generation and SDRAM control.

2.3.2 Other Components

A few other components deserve a mention

- **PLL settings**: The DSP is designed to operate at 225 MHz with a 50 MHz external oscillator.For normal operation on 6713 ,the internal phase lock loop is configured with s multiplier of 9 and a divider of 2 to obtain the 225 MHz clock from 50 MHz source. When the DSP comes out of reset ,the PLL is inactive and the system runs at 50MHz.the 6713 board support library initializes the PLL with the call DSK6713init().
- **Memory Overview**: The C6X family of DSPs have a single large 32 bit address space.The address space is split among on chip memory,external

memory and on chip peripheral registers .All memory is byte addressable and program code and data can be mixed freely. The C6713 has large 4K byte program and data caches to improve performance when accessing external code and data.

The compiler supports two memory models while accessing code and data. The near memory model is most efficient but handles only 32K bytes of data and 1 Mbyte of code.The far memory model has no such restrictions ,but requires additional instructions.

- **Endianess** : Endianess is a term referring to the byte ordering of multi byte data types.Specifically a system is called big endian if byte 0 contains the MSB or little Endian if byte 0 contains LSB. The . DSK default is little Endian.
- **Linker command files** : Code composer studio has several options of specifying where in memory your program should be placed. Linker command files ". cmd " are text files that give the location of code/data. Always ensure that your code is ending in a reasonable part of memory.
- **External Memory:** The 6713 DSK has two types of external memory
    1. 8Mbytes of synchronous DRAM : The DSK has a 64 Mbit SDRAM on the 32 bit EMIF. The SDRAM is mapped at the beginning of CE0 i.e. at the starting address (0x80000000).SDRAM must be periodically refreshed or it may lose memory
    2. 512 Kbytes of flash : Flash is a type of memory which doesn't lose its contents when the power is turned off. When read it lookes like asynchronous ROM. Flash can be erased and then each word can be reprogramed by proper commands
- **CODEC**:The DSK uses a Texas Instruments AIC23 (TLV320AIC23) stereo codec for input and output of audio signals.The codec samples analog signals on the microphone or line inputs and converts them into digital data that can be processed by the DSP. When the DSP is finished with the data it uses the codec to convert the samples back to analog signals on the line and headphone outputs so the user can hear the output.Preferred

CODEC setup is shown as

CONTROL CHANNEL –McBSP0,SPI mode with internal clocks in sync mode.

DATA CHANNEL-McBSP1,AIC23 in master mode with 12 MHz clock

- **LED INDICATORS** :The DSK has 8 LEDs made up of four status indicators and four user define LEDs. The user defined indicators are green LEDs next to DIP switches.

  User Leds The four ,user controllable LEDs allow for user feedback and display of simple status information. They are controlled by writing to the CPLD USER_REG register.They can also be set or cleared through the LED

  Module of Board support Library

  DIP SWITCHES The DIP switches allow simple feedback from the user. The DIP switches can be read from CPLD USER_REG register and the Board Support Library.

- **Emulation –USB and JTAG**
  - **USB :** Theembedded USB emulator is connected to your host PC through a 1.1 USB compatible interface. The emulator is responsible for regulation of debug traffic between your PC and JTAG. The debug is transparent and both your emulator and USB port can't be seen by DSP on your board.
  - **JTAG :** JTAG refers to a set of design rules that encompass testing,programing and debugging of chips. The JTAG interface allows an user to examine signals and registers on a chip through an external interface that only uses 5 extra pins on the chip

  Benefits of JTAG include

  - Your target hardware can be debugged as if it were a finished product. You don't need a special version of your hardware with debug features such as chip test clips or test points to expose signals.

- You don't need any special software running on your target hardware .Many systems use a ROM monitor to transfer debug information from host to target. But since the monitor itself uses resources like memory interrupt vectors and serial ports your pplication must be written within the ROM monitor.
- It is much faster and more flexible .You have all access to internal chip state at the same time,not just what your debug output happens to the monitor.

BSL : Theboard support library provides a C-language interface for configuring and controlling all on-board devices. The library consists of discrete modules that are built and archived into a library file. Each module represents an individual API and is referred to as API module.The module granularity is structured to provide a layer of hardware abstraction and software standardization that improves development time and modularity.

The BSL has 5 modules :

- Board Setup
- Codec
- DIP switch
- LED
- Flash

CHAPTER 3

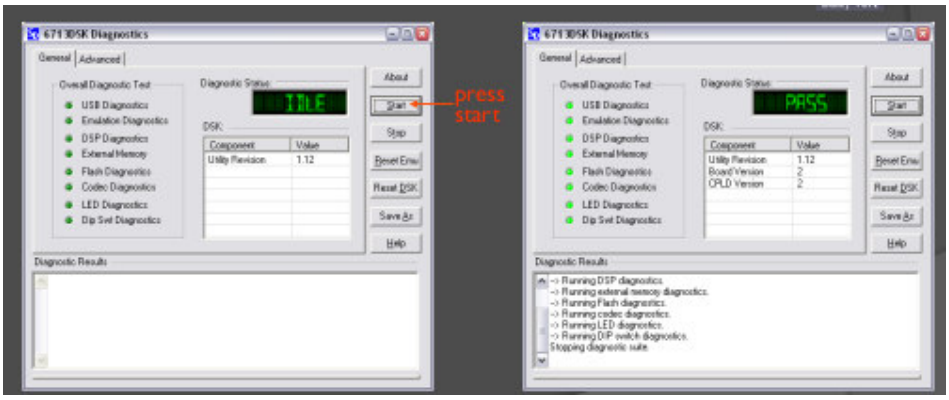
Programming with CCS

Everyone knows that debugging is twice as hard as writing a program in the first place. So if you are as clever as you can be when you write it, how will you ever debug it?

Brian Kernighan

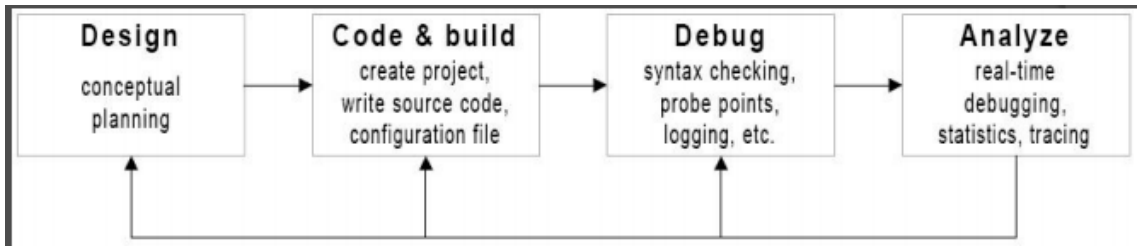
3.1 Post Test

We Power up DSK Board and watch LEDs till all are still. Power On Self Test (POST) program stored in FLASH memory automatically executes. This POST takes 10-15 seconds to complete. During POST, a 1kHz sinusoid is output from the AIC23 codec for 1 second. If POST is successful, all four LEDs blink 3 times and then remain on.

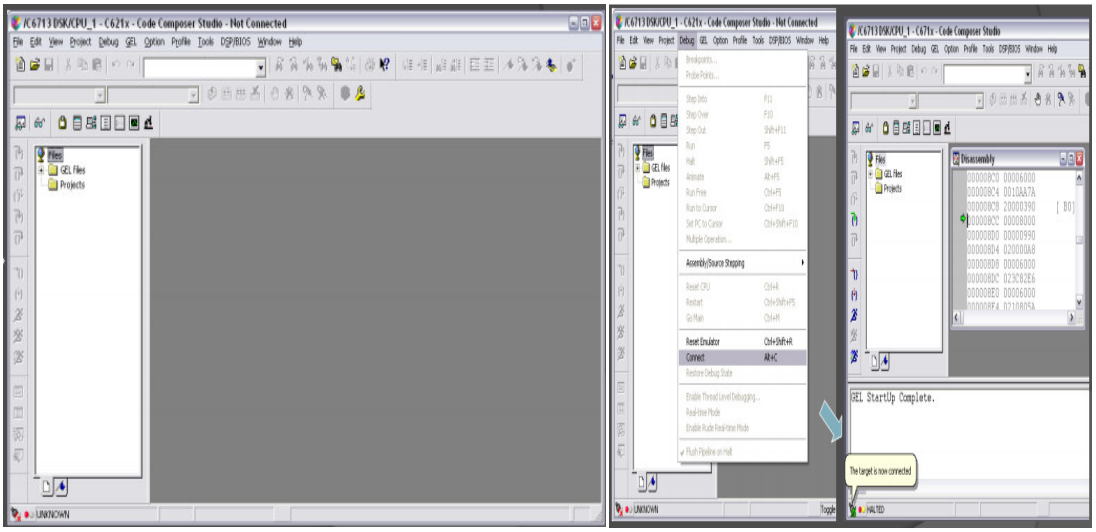


3.2 Open CCS

CCS or Code Composer studio by TI is the IDE we used for the Development Environment. The main features of it are given below



Feature of CCS



Open CCS and Press ALT+C to connect

3.3 Creation of a Project

We press **Ctrl+N or File→New Project**. We then give Location & Name to it and choose the target as **67XX**. We then create new source file from **File→New→Source File**. We save it in the projects directory. We can also add C files to the project.

3.4 Adding Libraries and Support Files

We add the following files to the project

C6713.cmd – Linker command file, we add from the support folder

rts6713.lib – Run time support Library from lib folder under C6000 folder

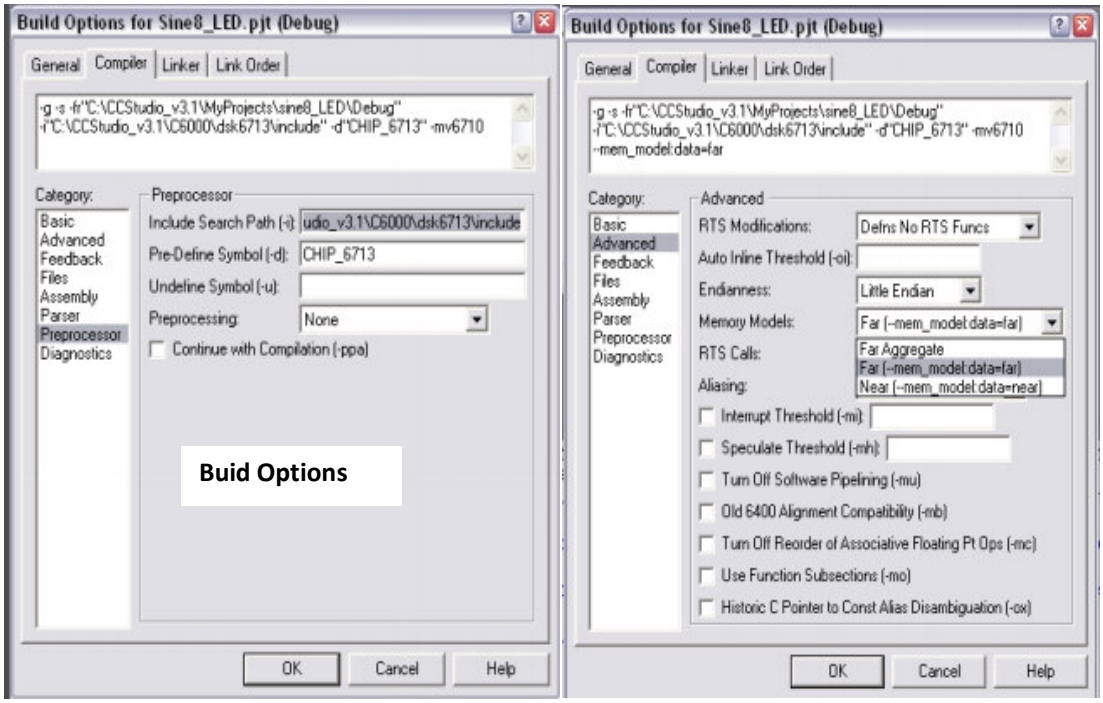
dsk6713 – DSK board support library from lib folder under C6000 folder

csl6713 – Chip support library from lib folder under C6000 folder

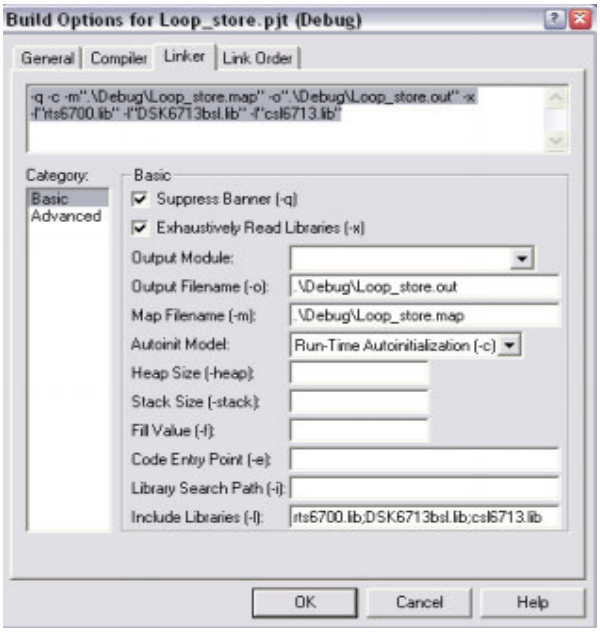
Vector_poll.asm or **Vector_interrupt.asm –** Interrupt as required

3.5 Setting the Build Options

Select Build Option under project menu. Change the Options as given below in the compiler tab. Change the memory model to far and rts call to are far



We change the option in the linker menu to accommodate the linker library files



Change the Linker as given in the fig

3.6 Compiling and Running the Program

Click **Build** from **Project** Menu. If error come the correct then rebuild all from project menu. After compiling a .out exec file will be created under a debug folder. It is dumped in the board by clicking **Load Program** from the **File** Menu. Else press ctrl+L to load the program on the board. Click Debug→Run to Run the program else click F5 to run. After running Click Halt from Debug Menu.

3.7 BSL Support

- To Initialize DIP/LEDs we use the function:

  DSK6713_DIP_init()/ DSK6713_LED_init()

- To Read state of DIP switches with

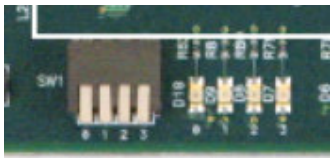
  DSK6713_DIP_get(n)

- To Change state of LEDs

  DSK6713_LED_on(n)

  DSK6713_LED_off(n)

  DSK6713_LED_toggle(n)

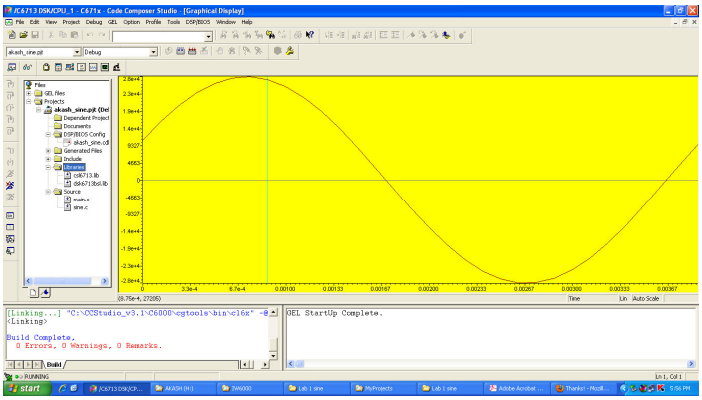


DIP SWITCHES AND LEDS

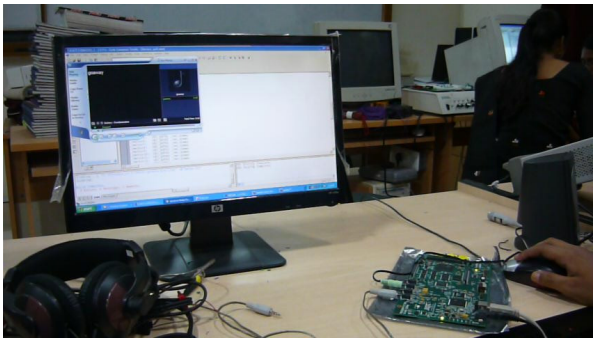
where n=0, 1, 2, or 3

- For AIC23 Codec we have McBSP. We write the ISR first.
  MCBSP_read() requests samples from the codec's ADC
  MCBSP_write()sends samples to the codec's DAC

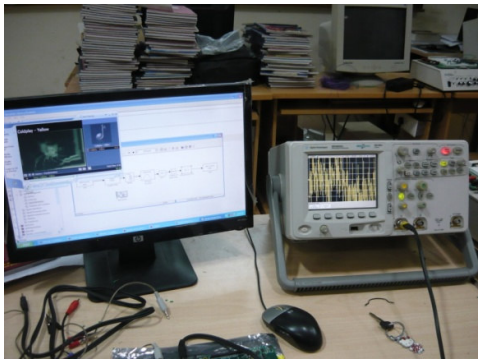
3.8 Experiments



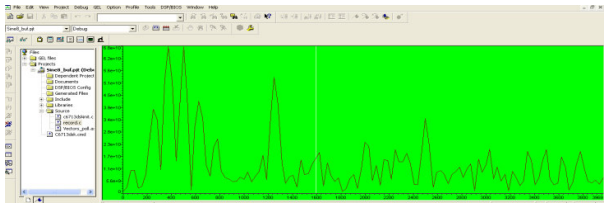
Sine Wave Generation



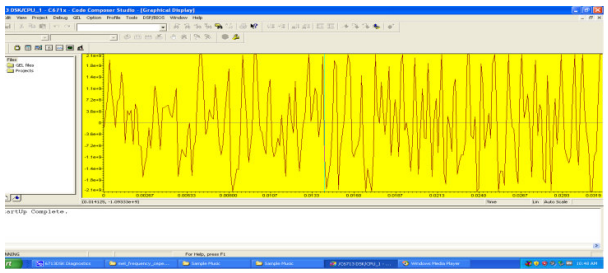
Echo on DSK



Audio Player on DSK



Freq Domain of Recorded Signal



Time Domain of Recorded Signal

CHAPTER 4

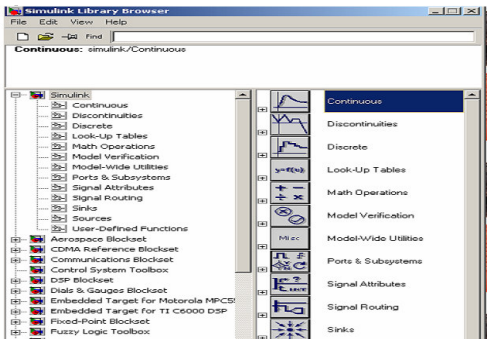
Simulink on TMS320C6713

You don't need to be a programming guru to use it

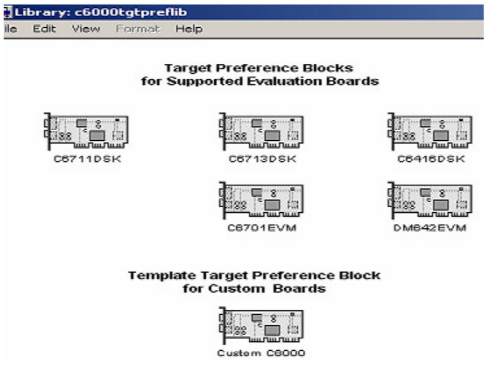
4.1 Introduction & Steps Involved

Embedded target for TIC6000 DSP platform integratesSimulinkwith DSPXpress tools. It allows one to generate DSP designs through code and prototype it on c6713 board. The build process builds a CCS project from the c code generated by real time windows target.It is automatically linked and compiled and ready to run on DSP.

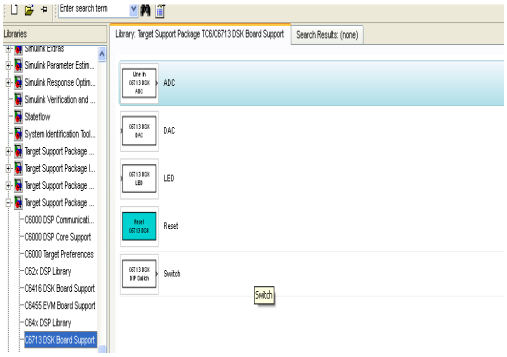
1) Open Simulink



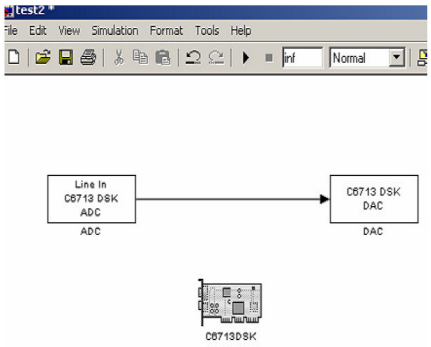
2) Open a new model and then select our target board from c6000tgtpreflib>>C6713 DSK as is shown below



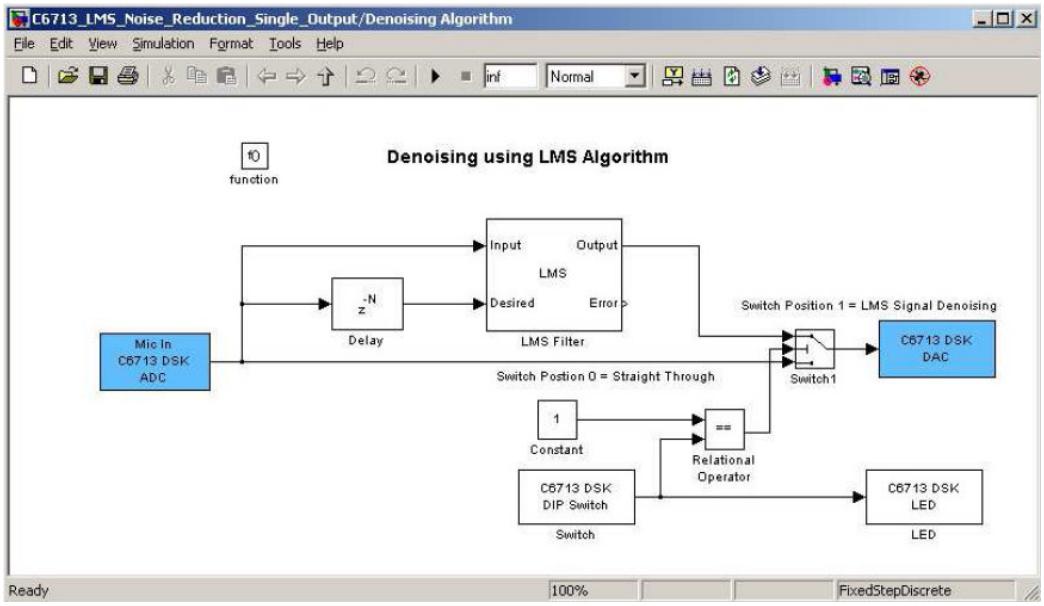
3) It is impossible to implement DSP on real time signal without adc, dac. Select them from C6713DSK Board support package from Tar get support package library.



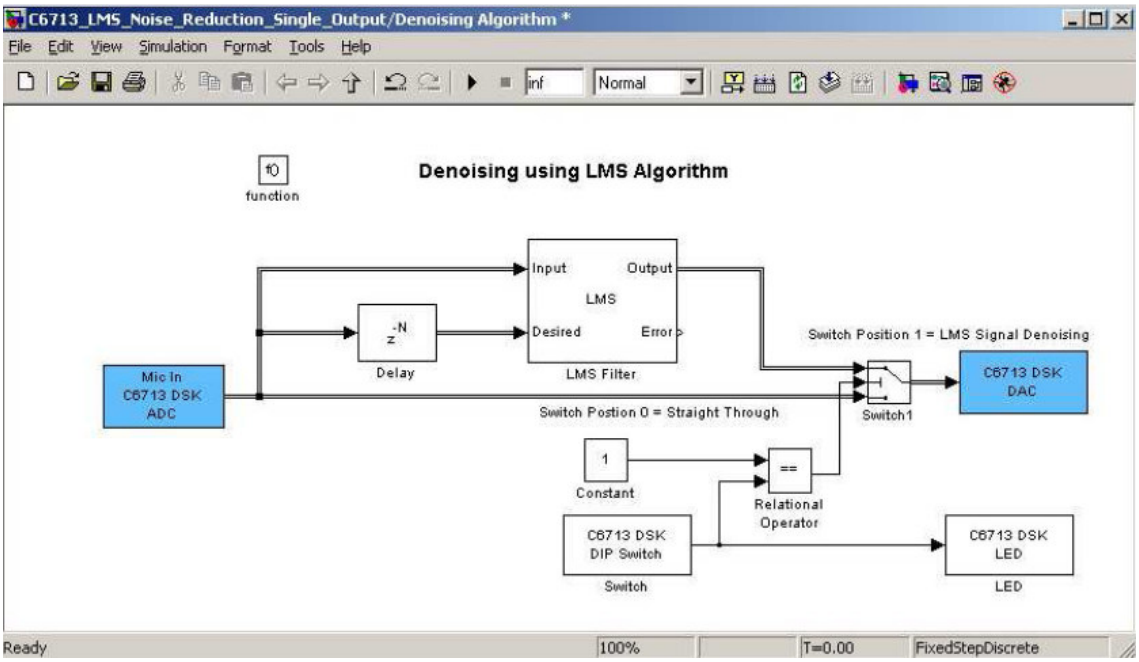
4) Compete your model using above target preferences, board support and Simulink toolboxes. Press ctrl+b to generate code and build



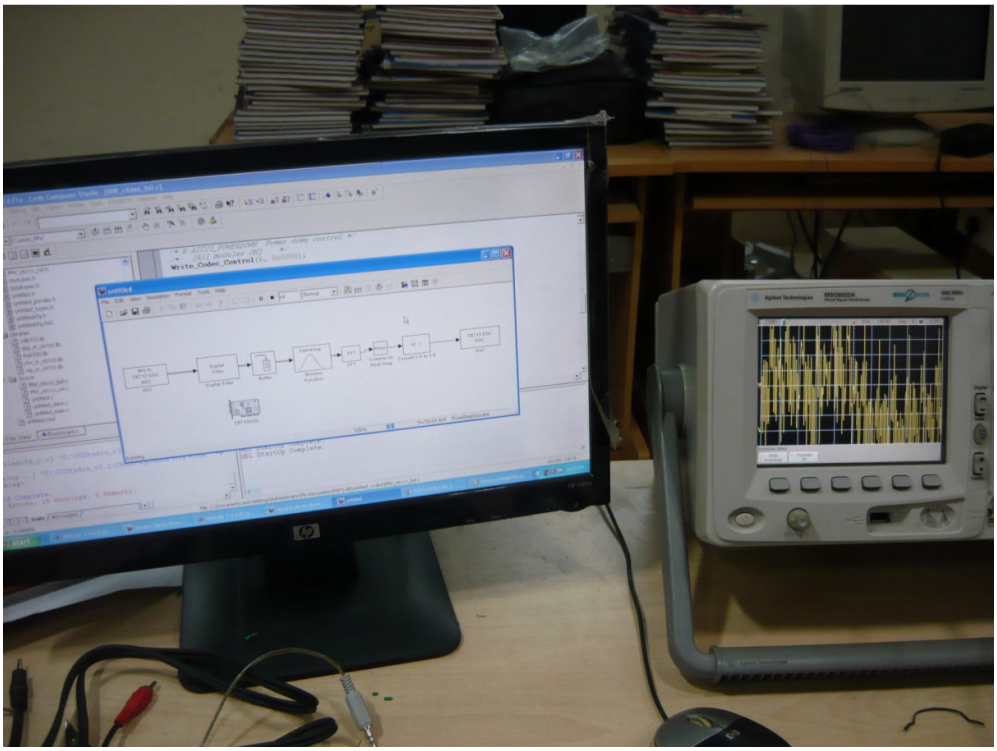
4.2 Simple Experiment on RT Speech



LMS on Simulink



Denoising using LMS



Real Time PreProcessing of audio Signals

CONCLUSION

The results obtained using Gaussian model mixing and VQ are almost perfect but there are still limitations. e.g.- if the number of speakers are greatly increased then we might not get so perfect results.

Also we couldn't implement the entire speaker recognition process i.e. the recognition part in CCS due to some complexities.So we could extend this project by using other algorithms for recognition like HMM and implementing the recognition part in CCS.

Hence we finally conclude though our project has limitations it has ousted its limitations.

REFERENCES

[1] Campbell ,J.P., Jr. , “Speaker recognition a tutorial ” Proceedings of the IEEE Volume 85,Issues 9, Sept. 1997 Pages:1437 – 1462

[2] Seddik, H. ;Rahmouni, A. ; Sayadi ,M. ; “Text independent speaker recognition using the Mel frequency Cepstral coefficients and a neural network classifier” First International Symposium on Control, Communications and Signal processing”, Proceedings of IEEE 2004

[3] Claudio Becchetti and LucioPrinaRicotti, “Speech Recognition”, Chichester: John
Wiley & Sons, 2004.

[4] Martin, A. and Przybocki, M., **“**The NIST Speaker Recognition Evaluations: 1996-
2000**”**, Proc. OdysseyWorkshop, Crete, June 2001

[5] Martin, A. and Przybocki, M., **“**The NIST 1999 Speaker Recognition Evaluation—An
Overview**”**, Digital Signal Processing, Vol. 10, Num. 1-3. January/April/July 2000

[6]Nakai, M.; Shimodaira, H.; Kimura, M.; “A fast VQ codebook design algorithm for a
large number of data”, IEEE International Conference on Acoustics, Speech, and Signal
Processing, Volume 1, Page(s):109 – 112, March 1992.

[7] S. Furui, “Speaker independent isolated word recognition using dynamic features of
speech spectrum”, IEEE Transactions on Acoustic, Speech, Signal Processing, Vol.34,
issue 1, Feb 1986, pp. 52-59.

[8]R. Chassaing, DSP Applications Using C and the TMS320C6x DSK, Wiley, New
York, 2002.

[9] Nasser Kehtarnavaz ,Real-Time Digital Signal Processing: Based on the Tms320C6000,Elsevier Inc. ,2005