International Journal of Modern Trends in Engineering and Research
www.ijmter.com
e-ISSN No.:2349-9745, Date: 2-4 July, 2015

# Vocabulary Speech Recognition System Using Word-based CHMM

Prof. Shweta k. Patil[1], Dr. D.M. Yadav[2]
[1]*Electronics & Tele-comm., JSPM'S Bhivarabai Sawant Institute Of Technology &Research Wagholi , Pune.*
patilshweta162@gmail.com
[2]*Electronics & Tele-comm., JSPM'S Rajarshi Shahu School Of research engineering Narhe , Pune.*
dineshyadav800@gmail.com

**Abstract-**The speech recognition system used in embedded application requires very high recognition accuracy and low-power consumption which we can achieve by implementation of word based continuous hidden Markov models based speech recognition (CHMM) system. This is mid-size vocabulary the speech recognition system based on the proposed coprocessor is applicable for the (100–1000 words) recognition tasks. In this paper implementation of custom-designed coprocessor for output probability calculation (OPC) is introduced .It is the most computation-intensive processing step in CHMM-based speech recognition algorithms. A polynomial addition-based method is used to compute add-log so that we can reduce the power consumption and save hardware resource. Xilinx Spartan-3A DSP XC3SD3400A is used to implement and test the proposed coprocessor & implementation of entire speech recognition system is done using SAMSUNG S3C44b0X as the micro-controller to execute the rest of speech processing. Above speech recognition system is tested using a 358-state 3-mixture 27-feature 800-word HMM.

**Keywords**-Speech recognition, Continuous Hidden Markov Model, hardware implementation, FPGA, Coprocessor Design.

## I. INTRODUCTION

The speech signal has very large variability, so it is better to perform some feature extraction that would reduce that variability. In the feature extraction, it is very common to perform a frequency warping of the frequency axis after the spectral computation. As shown in fig.1 the speech recognition system is combination of front-end and pattern recognition. The front end part converts the acoustic speech signals into short-time feature vectors. In the next half part pattern recognition process takes place in which we can find out a word sequence that best matches the input sequence of feature vectors. This is based on a set of pre-trained statistical models like hidden markov model.
A speech recognition system starts with a preprocessing stage, which takes a speech waveform as its input, and extracts from it feature vectors or observations which represent the information required to perform recognition. This stage is efficiently performed by software. The next step in recognition is decoding, which is obtains several samples of each possible word sequence, convert each sample to the corresponding acoustic vector sequence and run OPC and Viterbi decoding for the given input observation sequence (feature vectors of input speech) to the set of known samples word-level statistical models called hidden Markov models(HMMs)[1]

 

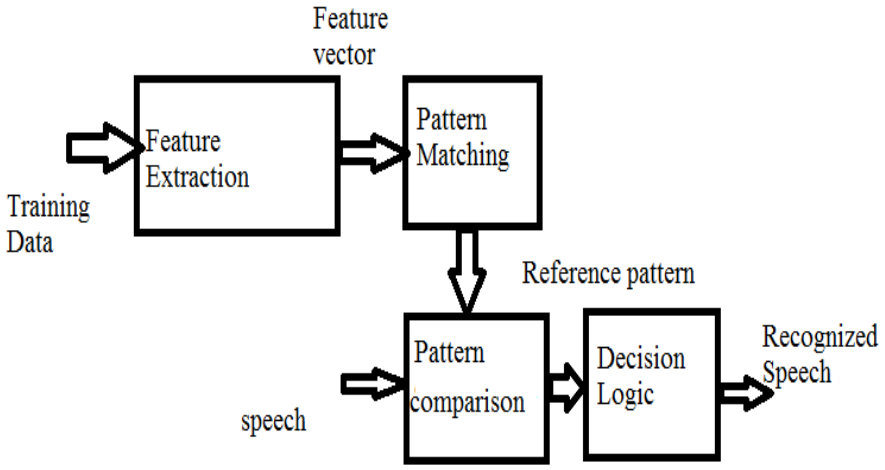


*Figure 1. Speech recognition system*

## II.CONTINUOUS HIDDEN MARKOV MODEL

From different types of HMM like discrete HMM (DHMM), semi-continuous HMM (SCHMM), and continuous HMM (CHMM), CHMM gives the best recognition rate at the price of more computation overhead .The output probability in CHMM is typically based a Gaussian distribution or a mixture of Gaussian distributions.

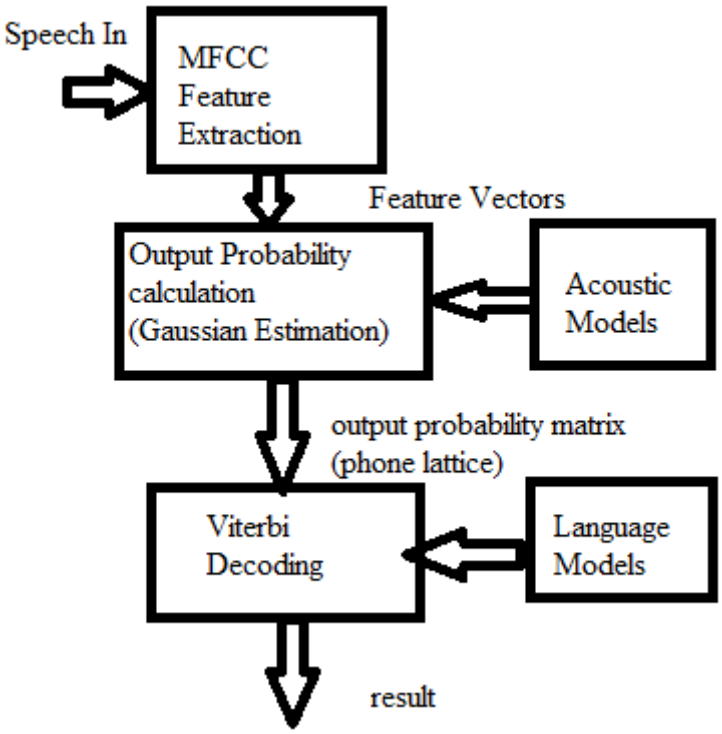


*Figure 2. Continuous hidden markov model*

 

CHMM contains three primary steps are MFCC feature extraction, output probability calculation, viterbi decoding. The speech input is typically recorded at sampling rate above 10 kHz. This sampling frequency was chosen to minimize the effects of aliasing in the analog-to-digital conversion. These sampled signals can capture all frequencies up to 5 kHz, which cover most energy of sounds that are generated by humans. As been discussed previously, the main purpose of the MFCC processor is to mimic the behavior of the human ears. In addition, rather than the speech waveforms themselves, MFFC's are shown to be less susceptible to mentioned variations.

**2.1 Mel-Frequency Cepstral Coefficients**

Fig.3 shows the MFCC processor in this frame blocking is the step where continuous speech signal is blocked into frames of N samples, with adjacent frames being separated by M (M < N). The first frame consists of the first N samples & second frame begins M samples. This process continues for all the speech is accounted for within one or more frames. Windowing is the next step in the processing is to minimize the signal discontinuities at the beginning and end of each frame windowing of each individual frame.The concept here is to minimize the spectral distortion by using the window to taper the signal to zero at the beginning and end of each frame. If we define the window as $w(n), 0 \le n \le N-1$, where N is the number of samples in each frame, then the result of windowing is the signal.

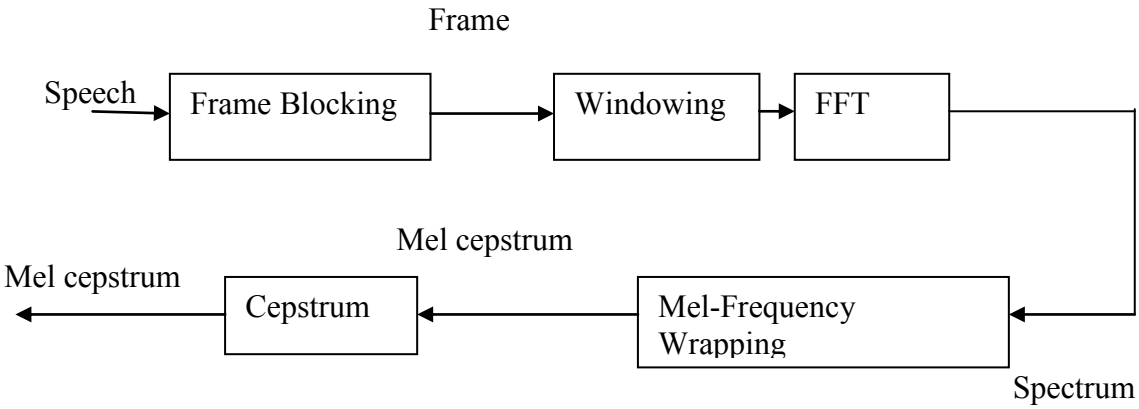


***Figure 3. Block diagram of the MFCC processor***

Fast Fourier Transform (FFT) is used to converts each frame of N samples from the time domain into the frequency domain. The result of this step is o referred to as spectrum or periodogram. In Mel-frequency Wrapping psychophysical studies have shown that human perception of the frequency contents of sounds for speech signals does not follow a linear scale. Thus for each tone with an actual frequency, f, we measure in Hz & a subjective pitch is measured on a scale called the 'mel' scale. The mel-frequency scale is linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. The final step in MFCC is Cepstrum here we convert the log Mel spectrum back to time & the result is called the Mel frequency cepstrum coefficients (MFCC).

 

### 2.2 Output Probability Calculation

Output Probability Calculation (OPC) is the most computation-intensive part of CHMM (Continuous Hidden Markov Model) based speech recognition algorithm .This paper presents a custom designed co-processor to implement OPC which help to reduce power consumption and design cost . To avoid underflow, a logarithmic representation is often used for all the probabilities. The output probability density function $b_j(O_t)$ in logarithmic domain ,here $O_t$ is the speech feature vector at frame t, $c_{jg}$ ,$\mu_{jg}$ and $\sum jg$ is the weight, the mean and the covariance matrix respectively for state j and the gth Gaussian mixture distribution. G denotes the number of mixture density functions and M denotes the dimension number of the feature vector.[2]

### 2.3 Viterbi Decoding

The Viterbi algorithm is used to find the likelihood of Gaussian mixture HMMs. Part of the Viterbi algorithm involves calculating a dot product between two vectors. The addition of the products may result in overflow if the representation of the values is not chosen properly.
The characteristics of how the chip handles each arithmetic operation were modeled using a C++ simulation of the Viterbi algorithm, the optimal bit shifts to apply at various places in the algorithm in order to avoid overflow. [3]

## III. VLSI IMPLEMENTATION OF COPROCESSOR

We design the speech recognition system which contains MCU and a coprocessor considering computation load for each processing step and flexibility of the speech recognition algorithm. The MFCC feature extraction, Viterbi decoding, and system control tasks this part comes under the MCU processes and the coprocessor takes care of OPC. The MCU in the formulate more flexibility system to adapt the different models and different recognition tasks. The proposed coprocessor for OPC could act as an SoC IP core. SRAM interface is interface between the MCU and the coprocessor, which makes the coprocessor be easily controlled by various MCUs, such as ARM and MIPS. [4]

### 3.1 Coprocessor Architecture

Fig. 3. Shows overall architecture of the proposed multi-PE coprocessor for OPC. PEs is the core processing blocks which used to compute OPC for states in parallel. Moreover, in each PE, Mahalanobis distance, and add-log calculation are computed in parallel. The key point that justifies this parallel processing is that the current add-log calculation is only related to the last Mahalanobis distance calculation with the same Gaussian mixture.
The speech feature vectors are stored in SRAM1, SRAM2, and SRAM3.The final OPC result from the PEs is store in the SRAM4 .To control the coprocessor MCU could directly configure the registers in the Register File. The registers have the same address space as SRAMs, which could be written by the MCU the same way as the SRAM.$d_{jg}$ and $A_p$ are also stored in Register File. The number of frames , the number of states , the number of Gaussian mixtures and the number of features defined are four HMM model parameters could be reconfigured in this unit. On the other hand, if we use the maximum capacities for the task, which is bits for *SRAM2* nd SRAM3 to store parameters of the state model for all states with mixtures and bits for SRAM1 to store the feature vectors of all frames, the total SRAM capacity would be nearly 488 kb, which occupy significant hardware resources and chip area.

 

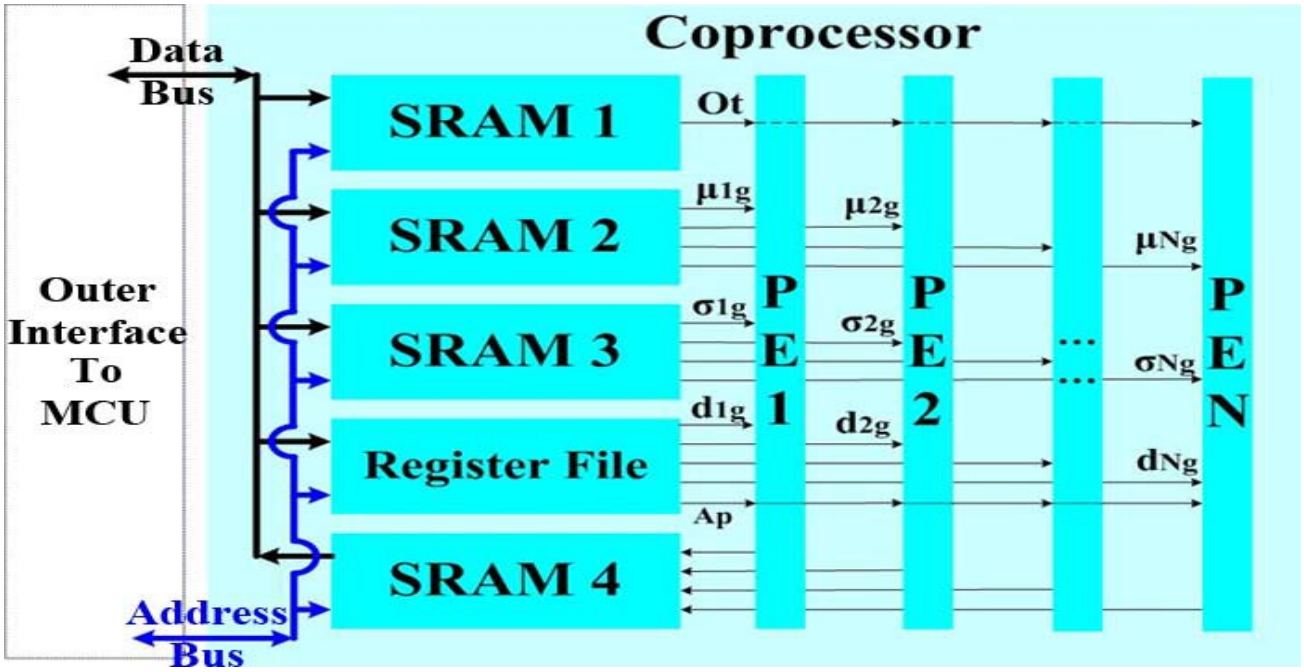


***Figure 4. Coprocessor Architecture.***

The architecture design shown in above fig. the size of SRAM2 and SRAM3 could be set to bits to allow parallel processing of the PEs. We need that feature vectors of all frames can be transferred once we set the size of SRAM1 to the above maximum size .A PE unit is composed of following three sub-units are MDC Unit, ALC Unit, and Interface Unit. In which MDC Unit performs a four-stage pipeline operation as subtraction, multiplication, square operation, and accumulation for Mahalanobis distance calculation .ALC Unit is used to calculate polynomial addition having a two-stage pipeline, multiplication and addition. The next unit is Interface Unit $Adder_1$ is used to calculate $Q_g$.[4]

Floating point and fixed-point C language both can be used for coding the speech recognition algorithm. The coprocessor is implemented on in Xilinx Spartan-3A DSPXC3SD3400A. Samsung S3C44b0x (containing an ARM7 core) is used as the MCU in speech recognition system. In this system, we use the same HMM model (358-state 3-mixture 27-feature 800-word HMM) is used to test the above speech recognition system.

.

## IV. CONCLUSION

CHMM based speech recognition algorithms have a very good recognition accuracy for word recognition tasks .To reduce power consumption and enhance flexibility, this paper presents a speech recognition system composed of a coprocessor and a MCU. The coprocessor is used to calculate Mahalanobis distance and add-log in parallel. Add-log calculation is based on polynomial fitting method is save the area as compared to look-up table method. Look –Up table-based approach is more speedy than the polynomial fitting-based approach to overcome this drawback we calculate the add-log and Mahalanobis distance in parallel. The word HMM-based recognition system requires a long processing time because in this system we calculate the likelihood scores for all reference models.

 

## REFERENCES

[1] S. J. Melnikoff, S. F. Quigley, and M. J. Russell, "Implementing simple continuous speech recognition System on an FPGA," in Proc. IEEE FCCM, 2002, pp. 275–276.
[2] P. Li and H. Tang, "Design a co-processor for output probability calculation in speech recognition," in Proc. IEEE ISCAS, 2009, pp. 369–372.
[3] E. Cornu, N. Destrez, A. Dufaux, H. Sheikhzadeh, and R. Brennan, "An ultra low power, ultra miniature Voice command system based on hidden Markov models," in IEEE Proc. ICASSP, 2002, vol. IV, pp . 3800–3803.
[4] Peng Li and Hua Tang, "Design of a Low-Power Coprocessor for Mid-Size Vocabulary Speech Recognition Systems" In IEEE Transactions On Circuits And Systems—I: Regular Papers, Vol. 58, No. 5, May 2011.

 