

# AN ULTRA LOW POWER, ULTRA MINIATURE VOICE COMMAND SYSTEM BASED ON HIDDEN MARKOV MODELS

*Etienne Cornu, Nicolas Destrez, Alain Dufaux, Hamid Sheikhzadeh, and Robert Brennan*

Dspfactory Ltd., 80 King Street South, Suite 206, Waterloo, Ontario, Canada N2J 1P5  
e-mail: robert.brennan@dspfactory.com

## ABSTRACT

A real-time HMM-based isolated word recognition system is implemented on an ultra low-power miniature DSP system. The DSP system consumes less than 1 milliWatt, much less than what is considered today as "low-resource". It has a very small footprint and requires only a single hearing aid sized 1 volt battery. The efficient implementation of HMM and MFCC feature extraction algorithms is accomplished through the use of three processing units running concurrently. In addition to the DSP core, an input/output processor creates frames of input speech signals, and a WOLA filterbank unit performs windowing, FFT and vector multiplications. A system evaluation using a vocabulary of 18 words shows a success rate of more than 99%.

## 1. INTRODUCTION

Speech recognition technology has recently reached a higher level of performance and robustness, allowing it to be deployed in a number of real-world environments, such as mobile phones and toys. As more applications are identified, the requirements for speech recognition algorithms also become more demanding: algorithms must run fast and use as little memory as possible so that they can be deployed in smaller and less expensive systems that use less space and less power. For example, Deligne et al [1] describe a low-resource continuous speech recognition system suitable for processors running at a minimum of 50 MIPS and having at least 1 MByte of memory, and Gong and Kao [2] describe a system running on a 30 MHz DSP with 64K words of memory. At the other end of the spectrum, J. Foks [3] presents a voice command system running on a 2.5 MHz CR16B processor and requiring only a few kilobytes of memory. The three systems are based on well-proven algorithms: all three use Mel Frequency Cepstral Coefficients (MFCC) to parameterize the input speech. For pattern matching, the first two use Hidden Markov Models (HMMs) and the

third uses Dynamic Time Warping (DTW). In contrast, Phipps and King [4] describe a voice command system based on Time Encoded Signal Processing and Recognition (TESPAR) that inherently requires much less processing power than MFCC extraction, DTW and HMM algorithms. It runs on an 8-bit 30 MHz 8051-type processor with less than 5 KBytes of memory. This type of processor typically consumes between 10 and 50 milliWatts of power.

This paper presents an HMM and MFCC-based voice command system comparable in functionality with low-resource systems such as the DTW and TESPAR-based systems mentioned above. However, it uses much less power due to the use of a DSP architecture designed specifically for speech processing in ultra low-resource environments. Consuming less than 1 milliWatt of power, the DSP system can run continuously for up to 100 hours, and whereas today's low-resource processors typically require AA batteries, the DSP system operates on a single hearing-aid sized battery, which is smaller than a penny. This allows voice command systems to be deployed in objects much smaller than before.

In the following sections, we first present an overview of the DSP hardware and describe how voice commands algorithms are mapped to the hardware components. We then describe how feature extraction, word endpoint detection and word likelihood calculations are performed on the system. The results of an evaluation performed using a specific configuration of the system are then presented, followed by a conclusion and a description of the work that will be done in the future.

## 2. THE DSP SYSTEM

The DSP system is implemented on two ASICs: a digital chip on 0.18 $\mu$  CMOS technology contains the DSP core, RAM, the weighted overlap-add (WOLA) filterbank, and the input-output processor (IOP). The mixed-signal portions are implemented on 1  $\mu$ m CMOS. A separate off-the-shelf E<sup>2</sup>PROM provides the non-volatile storage. The

RAM consists of two 4K-word data spaces and a 12K-word program memory space. Additional shared memory for the WOLA filterbank and the IOP is also provided. The core provides 1 MIPS/MHz operation and has a maximum clock rate of 4 MHz at 1 volt. At 1.8 volts, 30 MHz operation is also possible. The entire system operates on a single battery down to 0.9 volts and consumes less than 1 milliWatt. Prototype versions of the chipset are packaged into a 6.5 x 3.5 x 2.5 mm hybrid circuit.

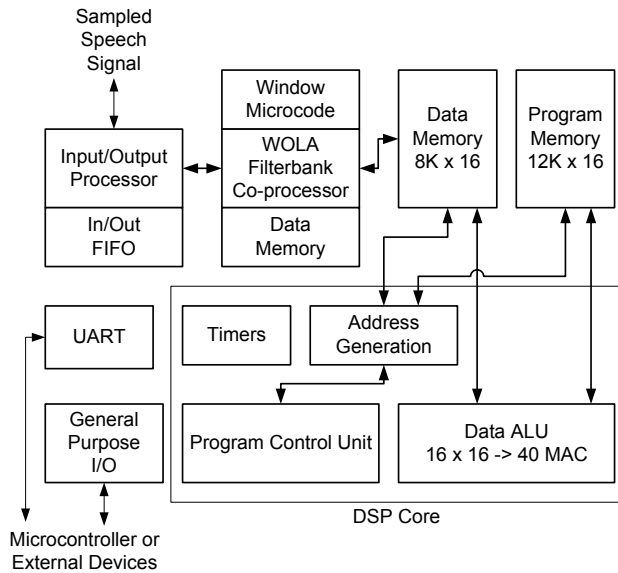


Figure 1 - DSP Block Diagram

Figure 1 shows a block diagram of the DSP [5]. The DSP communicates with the outside world through a UART (serial port), 16 general-purpose input/output pins and a channel dedicated to the speech signal coming from the mixed-signal chip. The 16 I/O pins can, of course, be used regardless of a whether a microcontroller is available or not. They have been used in the following functions:

- Input. They can be connected to switches to allow commands to be sent to the DSP system.
- Visual output. They can be connected to LEDs to inform the user of the current state of the system (training mode, recognition mode, etc.).
- Action output. They can be connected to various output devices. When the system recognizes a word, it can activate one or a combination of these pins to drive an external device, such as a speech synthesizer or a lamp.

Figure 2 illustrates the breakdown of the work between the three processors for the major voice command algorithm operations. The top five operations

are parts of the feature extraction and endpoint detection processes. The data produced by these processes is stored in a circular buffer where it is retrieved during the training and the recognition phases.

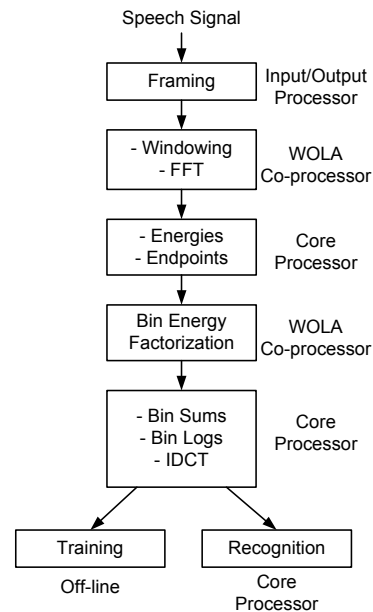


Figure 2 - Work Breakdown

### 3. FEATURE EXTRACTION

The input-output processor (IOP) is responsible for management of incoming and outgoing samples. In the voice command application, it takes as input the speech signal sampled by the 14-bit A/D converter on the mixed-signal chip at a frequency of 8 kHz. It creates frames of 256 samples, representing 32 milliseconds of speech. The frames overlap for 128 samples (16 milliseconds). A Hanning window is applied to each frame before it is made available to the core and the WOLA co-processor for processing.

The features most commonly used today in speech recognition systems are MFCCs and their first and second order differences. The number of coefficients and differences varies depending on the implementation; speech recognition systems running on fast processors typically use 12 or more coefficients and their first and second order differences for optimum recognition performance. The storage requirements for each word in the recognition vocabulary and the processing requirements are directly linked with the number of coefficients. Thus, this number has to be optimized based on the desired vocabulary size, response time and expected quality of the recognition.

Figure 3 illustrates how the different steps of feature extraction are performed on the DSP system. The three columns describe the tasks performed by the three processors running in parallel. The blocks in bold indicate the operations performed sequentially on a single 256-sample frame of data at the various stages of feature extraction. The blocks with dashed borders indicate the operations performed on the previous and next frames.

The MFCC calculation is launched when the input-output processor indicates that a new 256-sample frame is available for processing. This triggers a 256-point FFT on the WOLA co-processor. No data movement between the processors is necessary because the data resides in shared memory. When the 256-point FFT is complete, the DSP core determines the absolute value of each one of the 129 FFT bands as well as the total frame energy.

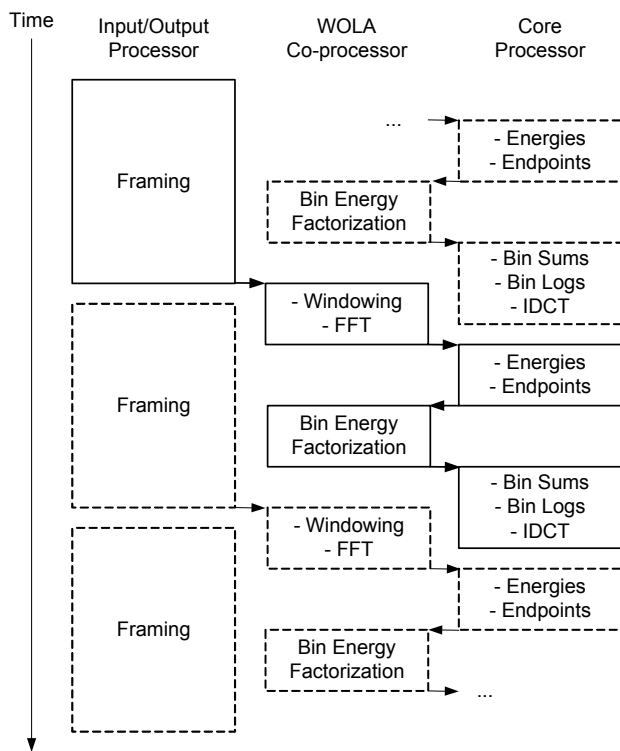


Figure 3 - Feature Extraction Task Assignment

The next step in the MFCC calculation consists in determining the log of the energy of L frequency bins, which are triangular bands spread non-linearly along the frequency axis. To do this, the DSP core launches the vector multiply function of WOLA co-processor, which multiplies the 129 FFT band energies by a vector of constants stored in RAM. When this operation is complete, the DSP core assigns the resulting values to the L frequency bins using a constant index table mapping the

FFT bands to the frequency bin. Finally, the log of these L values is taken using a base-2 log function included in the on-chip math library. The function uses a 32-point look-up table, executes in 9 cycles and has  $\pm 3\%$  accuracy.

The final step consists in calculating the Inverse Discrete Cosine Transform (IDCT) of the L log energy bins. The IDCT operation is implemented as the multiplication of the L log energy bins by a constant matrix, whose dimensions are L by the desired number of MFCC coefficients. Included in all matrix entries is a bit-shifting factor that prevents a sum overflow. Once calculated, the MFCC coefficients are stored in a circular buffer where they can be retrieved for training or recognition.

#### 4. ENDPOINT DETECTION

Given the real-time needs of the system and the limited memory resources available, an endpoint detection algorithm based on energy thresholds was chosen. The algorithm is executed by the DSP core in parallel with the feature extraction function after the total frame energy is computed. The energy thresholds are regularly updated as function of a noise floor that is calculated during silence frames.

#### 5. PATTERN MATCHING

The Viterbi algorithm is employed to find the likelihood of Gaussian mixture HMMs. One of the main difficulties encountered during the implementation was the fact that all model parameters, MFCC coefficients and temporary likelihoods maintained during the execution of the Viterbi algorithm had to be represented as 16-bit fixed-point values. There are three major issues linked with this representation:

1. The fixed-point data format in which each value is represented must be chosen in a way such as to minimize the loss of information during calculations. Model parameters, MFCC coefficients and log likelihoods all have a different dynamic range.
2. Information is lost during multiplications because the result must be truncated to 16 bits. The chip features a rounding instruction that reduces these quantization errors.
3. 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 in a C++ simulation of the Viterbi algorithm. A study was then performed to determine an optimal way of representing model parameters, MFCC coefficients and temporary likelihoods as 16-bit fixed-point numbers. The study also produced the optimal bit shifts to apply at various places in the algorithm in order to avoid overflow.

## 6. SYSTEM REALIZATION AND RESULTS

As mentioned earlier, feature vectors and HMMs can be customized based on the final application. In order to determine the characteristics of the system in terms of memory usage, processing requirements and recognition quality, we have performed an evaluation of the system using a sample configuration and a corpus recorded in a quiet office environment. MFCCs up to the 8<sup>th</sup> coefficient were used, the number of states was set to 4, and for each state a single Gaussian mixture with a diagonal covariance matrix was used. In this configuration, the HMM model representing each word requires only 82 words of memory. Given that about 4K words are available for word models, the system is capable of handling a vocabulary of about 50 words. Measurements performed using the DSP's timer indicate that the likelihood estimation for one model given the above configuration takes about 1000 CPU cycles per frame of input speech. At a CPU clock frequency of 1.28 MHz, the likelihood estimation for a vocabulary word takes on average 26 milliseconds.

The training and the recognition phase were both performed on-line and in real-time using a PC application that played recorded sound files through the sound card connected to the voice command system's audio input. For training, feature vectors calculated by the DSP system were retrieved by the PC application and the models calculated using a MATLAB application. The resulting models were then loaded to the voice command system for the recognition phase.

System evaluation was performed on a vocabulary of 18 English words that included the 10 digits and 8 commands. The corpus contained 68 instances of each word, for a total of 1224. The tests were performed using the cross-validation technique; that is, a number of iterations were executed in which the corpus was split randomly into a training set and a recognition set. The results showed an average recognition rate of 99.5% over 50 cross-validation iterations.

## 7. CONCLUSIONS AND FUTURE WORK

This work has shown that voice command systems based on HMMs can be successfully deployed on DSP systems that are much smaller and use much less power than ever before. The system presented in this paper uses less than 1 milliwatt; it is packaged in a 6.5 x 3.5 x 2.5 mm hybrid circuit and uses a very small hearing-aid type battery. Because the system is configurable in terms of features and HMM model characteristics, it will be able to support a large number of applications where HMMs are known to provide good results, such as speaker-independent voice command and speaker identification. For these applications, the characterization that we have performed will allow us to foresee the capabilities of the system in terms of latency, vocabulary size and accuracy.

Because the DSP system was specifically designed for speech processing applications, it is also very well suited for noise reduction, speech enhancement and voice activity detection algorithms. We intend to deploy these algorithms, either on the same DSP as the voice command system or on a second DSP running in parallel, in order to produce robustness adapted to the environment in which the voice command application will be deployed.

## 8. REFERENCES

- [1] S. Deligne et al., "Low-Resource Speech Recognition of 500-Word Vocabularies". *Proc. Eurospeech 2001*, pp. 1829-1832
- [2] Y. Gong and U.-H. Kao, "Implementing a high accuracy speaker-independent continuous speech recognizer on a fixed DSP". *Proc. ICASSP 2000*, pp. 3686-3689.
- [3] J. Foks. "Implementation of Speech Recognition on CR16B CompactRisc". *Proc. ICSPAT 2000*.
- [4] T.C. Phipps and R. A. King. "A Low-Power, Low-Complexity, Low-Cost TESPAS-Based Architecture For The Real-Time Classification Of Speech And Other Band-Limited Signals". *Proc. ICSPAT 2000*.
- [5] R. Brennan and T. Schneider, "A Flexible Filterbank Structure for Extensive Signal Manipulations in Digital Hearing Aids", *Proc. IEEE Int. Symp. Circuits and Systems*, pp.569-572, 1998.