A LOW-RESOURCE, MINIATURE IMPLEMENTATION OF THE ETSI DISTRIBUTED SPEECH RECOGNITION FRONT-END

Etienne Cornu, Hamid Sheikhzadeh

Dspfactory Ltd., 611 Kumpf Drive, Unit 200, Waterloo, Ontario, Canada N2V 1K8

ABSTRACT

The purpose of this work is to demonstrate that distributed speech recognition front-ends can be deployed in environments which provide for very little power and CPU resources, with possibly no degradation of speech recognition quality when compared to standard floating-point implementations. The ETSI distributed speech recognition front-end standard is implemented on an ultra low-power miniature DSP system. The efficient implementation of the ETSI algorithm components, i.e. feature extraction, feature compression and multi-framing, is accomplished through the use of three processing units running concurrently. In addition to a DSP core, an input/output processor creates frames of input speech signals, and a weighted overlap-add (WOLA) filterbank unit performs windowing, FFT and vector multiplications. System evaluation using the TI digits database shows that the performance of the ultra low-power DSP system is equivalent to the reference implementation provided by ETSI.

1. INTRODUCTION

An increasing number of voice recognition based services are becoming available for mobile users. Due to the limited resources available on mobile devices, speech is transmitted to a server that performs the voice recognition task. One of the problems encountered in this approach is that artifacts are introduced in the speech through the use of codecs. Although these artifacts do not affect speech intelligibility, they have a definite impact on speech recognition performance. Distributed Speech Recognition (DSR) deals with this problem by moving the front-end of the voice recognition task to the mobile device. In April 2000, the Aurora DSR Working Group within ETSI published a standard (see [1] and [2]) that describes a set of algorithms for extracting features from speech and transmitting them in digital format to the server where voice recognition is performed.

The ETSI feature extraction standard is based on Mel Frequency Cesptrum Coefficients (MFCCs); it involves a number of computationally heavy operations such as FFT, energy calculation, frequency bin calculation, Inverse Discrete Cosine Transform (IDCT) and vector quantization. Given the complexity of these operations, it may prove difficult to deploy the standard in environments where only limited power and CPU cycles are available unless a solution specifically adapted to low-resource conditions is used.

This paper describes the implementation of the complete ETSI DSR front-end standard, i.e. feature extraction, feature compression and multi-framing, on a DSP system designed specifically for speech processing in ultra low-resource environments. Consuming less than 1 milliWatt of power, the DSP system can run continuously while hardly affecting the battery life of mobile devices. Therefore, the integration of the DSP system in mobile phones, PDAs, headsets and other mobile devices should allow an increasing number of voice-based applications to be deployed in the future.

In the following sections, we first present an overview of the DSP hardware and describe how the tasks of the ETSI standard are mapped to the hardware components. We then describe the specifics of how feature extraction, feature compression and multi-framing are performed on the system. The results of an evaluation performed using the TI digits 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 0.18μ CMOS technology contains the DSP core, RAM, the weighted overlap-add (WOLA) Co-processor, and the input-output Processor (IOP). The mixed-signal portions are implemented on a 1 μm CMOS chip. A separate off-the-shelf E^2PROM 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 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 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 miniature MCM package.

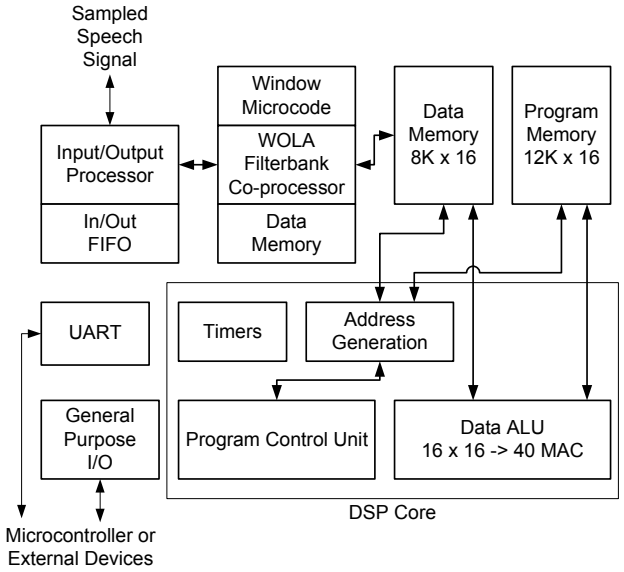

Figure 1: DSP Block Diagram.

Figure 1 shows a block diagram of the DSP (See [3] for additional information). Figure 2 illustrates the different operations that are part of the ETSI front-end and how they map to the different processing units in the DSP system. The top six blocks represent the feature extraction algorithm, the seventh represents the feature compression algorithm and the last two blocks represent framing, bit-stream formatting and error protection. The implementation of the three algorithms is described in the following sections.

3. FEATURE EXTRACTION

The input-output processor (IOP) is responsible for management of incoming and outgoing samples. 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. The mixed-signal chip also applies a DC offset filter to the speech signal. The IOP creates frames of 200 samples, representing 25 milliseconds of speech. The frames overlap for 80 samples (10 milliseconds).

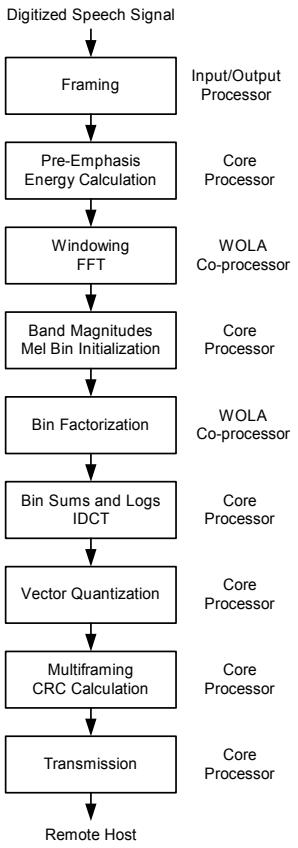

Figure 2: Work Breakdown

The MFCC and energy calculation is launched when the input-output processor indicates that 80 new samples are available for processing. The Core Processor first determines the energy of the new block of samples and applies a pre-emphasis filter. The WOLA co-processor then applies a 200-point Hamming window and performs a zero-padded 256-point FFT. No data movement between the three processors is necessary at this stage because the data resides in shared memory.

When the FFT is complete, the next step in the MFCC calculation consists in determining the log of the energy of 23 frequency bins, which are triangular bands spread non-linearly along the frequency axis. To do this, the DSP core first calculates the energy magnitude in each FFT band and initializes each frequency bin with the energy of the band situated in the middle of the bin. With a few exceptions, each 128 FFT band contributes to two frequency bins and therefore must be multiplied by two constants between 0 and 1. This multiplication of the 128

FFT band energies by two constants each is performed more efficiently by the WOLA co-processor. Here again, no data movement is necessary between the processors since all data resides in shared memory.

When the multiplication is complete, the DSP core assigns the resulting values to the 23 frequency bins using a constant index table mapping the FFT bands to the frequency bins. Finally, the log of these 23 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 ± 3% accuracy.

The final step of feature extraction consists in calculating the IDCT of the 23 log energy bins. The IDCT operation is implemented as the multiplication of the 23 log energy bins by a constant matrix consisting of 23x12 coefficients. Included in all matrix entries is a bit-shifting factor that prevents a sum overflow. Taking advantage of the Core Processor's single-cycle multiply-and-accumulate instruction, the IDCT is performed using only 360 processor cycles (about 0.28 milliseconds).

4. FEATURE COMPRESSION

The ETSI standard makes use of vector quantization (VQ) to compress features as seven pairs of two features. A codebook of 64 entries is used for MFCCs 1 through 12, and a codebook of 256 entries is used for MFCC coefficient 0 and the log energy. In order to make the quantization process more efficient for the DSP system, a sub-optimal MxL VQ codebook was trained for each feature pair as follows. First, M codewords were trained on the whole feature space. Next, for each of the M codewords, only the features already associated with that particular codeword were used to train L sub-codewords (see [4]). The first six codebooks use 8x8 codebooks and the seventh codebook is arranged as a 16x16 codebook. This significantly reduces the number of distance calculations that have to be performed, from 64 to 16 for the first six pairs and from 256 to 32 for the seventh pair. Off-line simulations showed no performance degradation due to the use of the sub-optimal VQ codebooks instead of the optimal codebooks suggested by the ETSI standard.

Despite this optimization, the DSP system has only enough time to perform the first six codebook searches before the IO Processor makes the next frame available, which occurs exactly every 10 milliseconds. Fortunately, the seventh codebook search can be delayed by a millisecond and performed by the DSP Core Processor while the WOLA Co-processor performs windowing and the FFT on the next speech frame, as illustrated in figure 3. The three columns show the tasks performed by the three processors running in parallel.

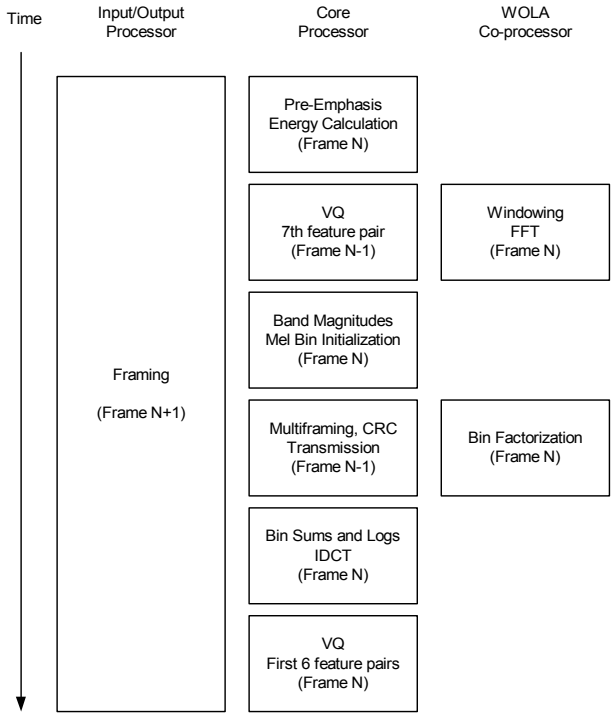

Figure 3: Operations performed during each 10 millisecond time period

5. MULTI-FRAMING

Before the compressed features are transmitted to a host, they are packed into 44 bits (5.5 bytes) and a 4-bit CRC is added every two frames. A 6-byte multi-frame header is also sent to the host before each set of 24 frames. This multi-frame header contains information such as a counter, the sampling frequency and a 16-bit CRC. These operations are all performed by the DSP Core while the WOLA Co-processor is busy multiplying the FFT bands as part of the feature extraction process for the next speech frame, as illustrated in figure 3.

6. EVALUATION

ETSI provides a floating-point based 'C' reference implementation of the feature extraction algorithm (See [1]). Given that the employed DSP system uses 16-bit fixed-point arithmetic, some differences between the two implementations will exist. Rather than measuring these differences, which would have been difficult anyway since the DSP system's input is analog, we compared the performance of both feature extraction systems in a speaker-dependent, isolated word recognition task in which the word likelihood calculation phase was

performed by HTK (HMM toolkit). In fact, in [2] Pierce notes that applying vector quantization does not negatively affect the word error rate when compared to using the original features before quantization. By our observations, the errors introduced by using fixed-point arithmetic are relatively small compared to the quantization errors, and therefore intuitively they should have minimal impact.

To perform the isolated word recognition task, we randomly selected 4 male and 4 female speakers in the original clean TI digits database. For each speaker, the database contains up to 26 instances of each English digit. We first extracted the features for all words of all 8 speakers and combined them to train the vector quantization codebooks, as described in section 4: one set of codebooks for the reference implementation and one set for the DSP system. Each codebook was trained using approximately 400,000 feature vectors. The trained vector quantization codebooks were then used for quantizing the features.

We employed HTK for HMM training and testing using the quantized features extracted by the reference implementation. The same procedure was performed using the features extracted by the DSP system. For each of the 8 speakers, we performed 50 cross-validation iterations. During each iteration, 10 instances of each digit were randomly selected and included in the training set; the rest were assigned to the test set.

The word-error rate (WER) averaged over all speakers was 0.564% for the reference implementation and 0.586% for the DSP system. The WER difference of 0.022% can be judged insignificant, especially when we consider the large variance amongst speakers in both evaluations: for the reference implementation, the WERs ranged from 0.025% to 1.475% and for the DSP system they ranged from 0.000% to 1.443%. We can therefore conclude that there is no noticeable degradation of the quality of the features generated by our system when compared to the features generated by the reference implementation.

7. CONCLUSIONS AND FUTURE WORK

This work has shown the ETSI front-end for distributed speech recognition can be successfully deployed on DSP systems that are very small and use very little power. There is no noticeable degradation in performance when compared to the floating-point based reference implementation provided by ETSI. 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 miniature MCM package and runs off a very small 1.2 V battery. It is therefore well suited for deployment in devices such as headsets, PDAs and mobile phones where distributed speech recognition is used and where low power consumption and small footprint are crucial.

The current ETSI standard is geared towards environments with low background noise. The next version of the standard will apply to noisy environments such as cars. The next step of our work will be to implement this new standard when it becomes available. Since the new algorithms are bound to be more complex than the current one, we expect that the 1.28 MHz processor will not be fast enough. However, the system clock can be set higher until it reaches a point where the algorithm can be performed in real-time. It is expected that the new standard will also be based on MFCCs and therefore the current DSP system will also be suitable.

8. REFERENCES

[1] "ETSI ES 201 108 v1.1.2 Distributed Speech Recognition; Front–end Feature Extraction Algorithm; Compression Algorithm", April 2000.

[2] D. Pierce. "Enabling New Speech Driven Services for Mobile Devices: An overview of the ETSI standards activities for Distributed Speech Recognition Front-ends". *AVIOS 2000*.

[3] 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.

[4] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1992.