Energy-Efficient Floating-Point MFCC Extraction Architecture for Speech Recognition Systems

Jihyuck Jo, Hoyoung Yoo, and In-Cheol Park

***Abstract*— This brief presents an energy-efficient architecture to extract mel-frequency cepstrum coefficients (MFCCs) for real-time speech recognition systems. Based on the algorithmic property of MFCC feature extraction, the architecture is designed with floating-point arithmetic units to cover a wide dynamic range with a small bit-width. Moreover, various operations required in the MFCC extraction are examined to optimize operational bit-width and lookup tables needed to compute nonlinear functions, such as trigonometric and logarithmic functions. In addition, the dataflow of MFCC extraction is tailored to minimize the computation time. As a result, the energy consumption is considerably reduced compared with previous MFCC extraction systems.**

***Index Terms*— Floating-point operations, hardware optimization, mel-frequency cepstrum coefficients (MFCCs), speech recognition.**

I. Introduction

Among diverse human-device interfaces, speech recognition has widely been used in the last decade, and its importance becomes higher as the era of the Internet of Things comes close to reality [1]. Due to the prevalence of energy-limited devices, energy-efficient architecture is inevitably demanded to lengthen the device life. The demand for low-energy architecture leads to the speech recognition system being implemented with dedicated hardware units [2].

A speech recognition system consists of two processes: 1) feature extraction and 2) classification [3], [4]. The feature extraction process picks the characteristics of a sound frame, and a word is selected in the classification process by analyzing the extracted features. This brief mainly focuses on the hardware design of feature extraction.

The most widely known feature extraction is based on the mel-frequency cepstrum coefficients (MFCCs), as MFCC-based systems are usually associated with high recognition accuracy [5]. In [6], MFCC extraction was implemented with an optimized recognition program running on a low-power reduced instruction set computer processor platform. To reduce energy consumption further, dedicated architectures have been proposed in [7]–[9] and constructed with fixed-point operations. The previous architectures, however, have not fully considered the arithmetic property of the MFCC extraction algorithm.

This brief presents a new energy-efficient architecture for MFCC extraction. Investigating the algorithmic property of MFCC extraction, we renovate the previous architecture with optimization techniques to reduce both hardware complexity and computation time. As a result, the energy consumption is remarkably reduced compared with the previous architectures.

The rest of this brief is organized as follows. Section II explains the conventional MFCC extraction algorithm. Section III describes the modified MFCC extraction system and the optimization techniques. How the proposed architecture is designed is described in Section IV, and it is evaluated in Section V. Finally, the conclusion is drawn in Section VI.

Manuscript received September 29, 2014; revised January 16, 2015; accepted March 7, 2015. This work was supported by the Center for Integrated Smart Sensors through the Ministry of Science, ICT and Future Planning as Global Frontier under Project CISS-2011-0031860.

The authors are with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea (e-mail: jhjo.ics@gmail.com; hyyoo.ics@gmail.com; icpark@kaist.edu).

Digital Object Identifier 10.1109/TVLSI.2015.2413454

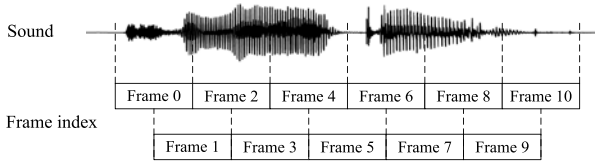

Fig. 1. Sound frame formation.

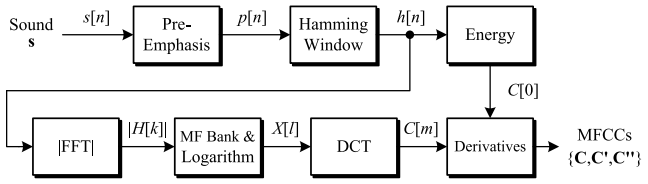

Fig. 2. Overall process for MFCC feature extraction.

II. Overview on MFCC Extraction Algorithm

MFCC extraction is a process that extracts features representing the characteristics of a sound frame. As shown in Fig. 1, a sound frame consists of N sound samples and it is half-overlapped in the time domain with the previous and next frames.

The overall flow of the conventional MFCC extraction is shown in Fig. 2, which extracts MFCC vectors for a sound frame $\mathbf{s} = \{s[0], s[1], \ldots, s[N-1]\}$. The MFCC vectors consist of $\{\mathbf{C}, \mathbf{C}', \mathbf{C}''\}$, where the first feature vector, $\mathbf{C} = \{C[0], C[1], \ldots, C[M-1]\}$, is a set of $(M-1)$ MFCCs and the logarithmic energy of the sound signals contained in the frame, and $\mathbf{C}'$ and $\mathbf{C}''$ indicate the first and second derivatives of $\mathbf{C}$, respectively. The rest of this section explains how MFCC vectors are computed in detail.

The given signal $s[n]$ is preprocessed by applying preemphasis and Hamming windowing consecutively. The underlining concept of the preemphasis is to amplify high-frequency components obtained by passing through a high-pass filter. The filtered output is given by

$$p[n] = s[n] - 0.97 \cdot s[n-1]. \tag{1}$$

Afterward, it is required to degrade $p[n]$ by multiplying the following Hamming window function in order to compensate the overlap between the neighboring frames. The degraded signal is computed as

$$h[n] = p\,[n] \cdot \left\{0.54 - 0.46 \cdot \cos\left(\frac{2\pi n}{N-1}\right)\right\}. \tag{2}$$

After the preemphasis and Hamming windowing are completed, the logarithmic energy of the sound frame is calculated as

$$C[0] = \log\left(\sum_{n=0}^{N-1} h^2[n]\right) \tag{3}$$

where $C[0]$ is the first component of $\mathbf{C}$.

In order to analyze $h[n]$ in the frequency domain, an N-point fast Fourier transform (FFT) is applied to convert them into their frequencies. The value and the magnitude of a frequency component resulting from the FFT can be expressed as

$$H[k] = \sum_{n=0}^{N-1} h(n) \cdot e^{-j\frac{2\pi n}{N}} \tag{4}$$

$$|H[k]| = \sqrt{(\mathrm{Re}(H[k]))^2 + (\mathrm{Im}(H[k]))^2}. \tag{5}$$

The logarithmic power spectrum on the mel-scale is computed using a filter bank consisting of L mel filters

$$X[l] = \log\left(\sum_{k=k_{ll}}^{k_{lu}} |H[k]| \cdot W_l[k]\right) \tag{6}$$

where $l = 0, 1, \ldots, L-1$; $W_l[k]$ is the lth triangular filter; and k_{ll} and k_{lu} are the lower and upper bounds of the lth filter, respectively. The lower and upper bounds of a filter are determined by considering the relationship between the frequency and the mel-scale [10].

The remaining MFCCs in $\mathbf{C}$ are obtained by applying the discrete cosine transform (DCT) to $X[l]$

$$C[m] = \sum_{l=1}^{L} X[l] \cos\left(\frac{\pi m(l-0.5)}{L}\right) \tag{7}$$

where $m = 1, \ldots, M-1$. The resulting features represent the frequency components of $X[l]$.

Let $\{\mathbf{C_i}, \mathbf{C_i'}, \mathbf{C_i''}\}$ be the MFCC vectors of the ith frame. Then, the two derivatives, $\mathbf{C_i'}$ and $\mathbf{C_i''}$, are calculated by

$$\mathbf{C_i'} = \mathbf{C_{i+2}} + \mathbf{C_{i+1}} - \mathbf{C_{i-1}} - \mathbf{C_{i-2}} \tag{8}$$

$$\mathbf{C_i''} = \mathbf{C'_{i+2}} + \mathbf{C'_{i+1}} - \mathbf{C'_{i-1}} - \mathbf{C'_{i-2}}. \tag{9}$$

Since $\mathbf{C_i}$ consists of M features, the total number of extracted features is $3M$ per frame.

III. Modified MFCC Extraction Algorithm

In this section, we first suggest a modified MFCC extraction algorithm, which is developed to make the implementation feasible in a low-cost circuit. The modification is achieved by preserving the recognition rate. Then, design parameters, such as the operation bit-width and the number of constant values, are determined by considering the algorithmic properties of the MFCC extraction process.

A. Modified MFCC Extraction

To process the entire MFCC extraction with simple operation units, the trigonometric function calculation is reconsidered, and the operation of the mel filtering is simplified by reusing some values computed previously.

In the trigonometric functions of (2), (4), and (7), the denominator in the phase is set to a power of 2 so that all the trigonometric functions can share a unified lookup table (LUT). In case of (2), N is used instead of $(N-1)$ in the denominator, and this slight modification results in almost no degradation of the recognition rate. Hence, the equation becomes

$$h[n] = p[n] \cdot \left\{0.54 - 0.46 \cdot \cos\left(\frac{2\pi n}{N}\right)\right\}. \tag{10}$$

In the mel filtering stage, the logarithmic power spectrum is calculated using a filter bank. As shown in Fig. 3(a), triangular filters are utilized conventionally [7]–[9]. However, it has been known that

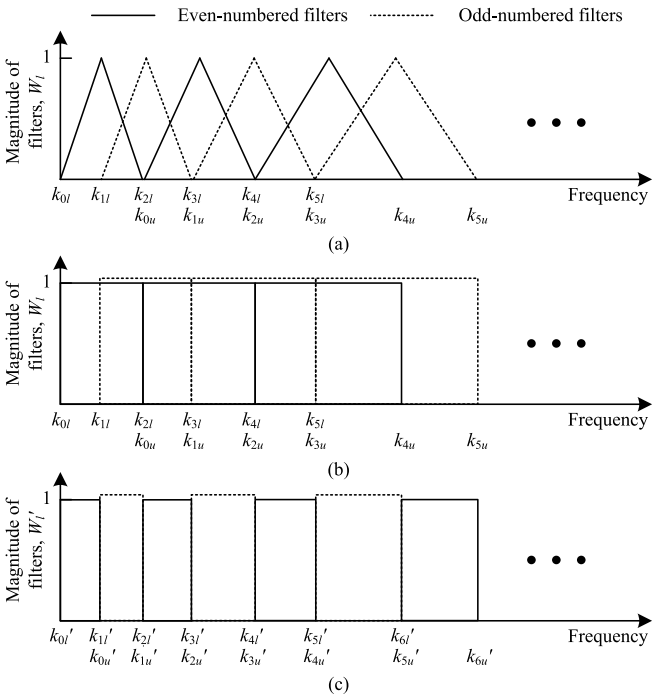

Fig. 3. Mel filter banks. (a) Triangular shape. (b) Full-sized rectangular shape. (c) Proposed half-sized rectangular shape (the magnitude of a rectangular filter is 1).

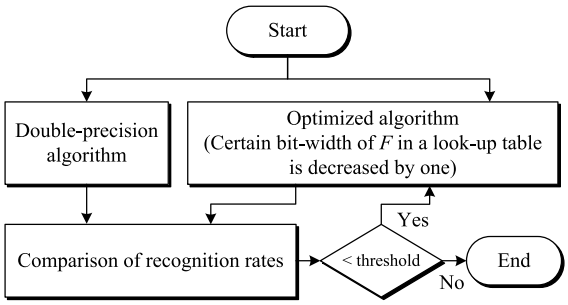

Fig. 4. Bit-width optimization for a lookup table.

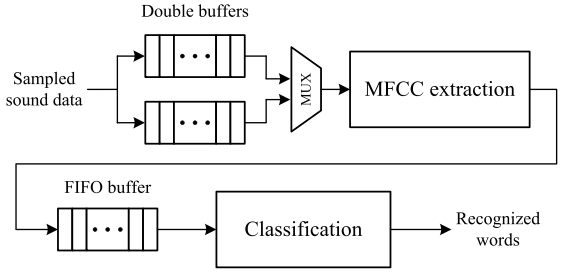

Fig. 5. Conceptual view of a speech recognition system.

the rectangular filter bank shown in Fig. 3(b) has negligible effects on the recognition rate [11]. In Fig. 3(b), the end point of an odd-numbed filter is equal to the start point of the next odd-numbed filter, and this relation holds for even-numbered filters. Based on this fact, we further simplify the filtering operation by considering the overlapped regions. As shown in Fig. 3(c), the proposed filtering is based on $W_l'[k]$ instead of $W_l[k]$, and $W_l'[k]$ is approximately half-sized compared with $W_l[k]$. Intermediate common subexpressions $Y[l]$ are

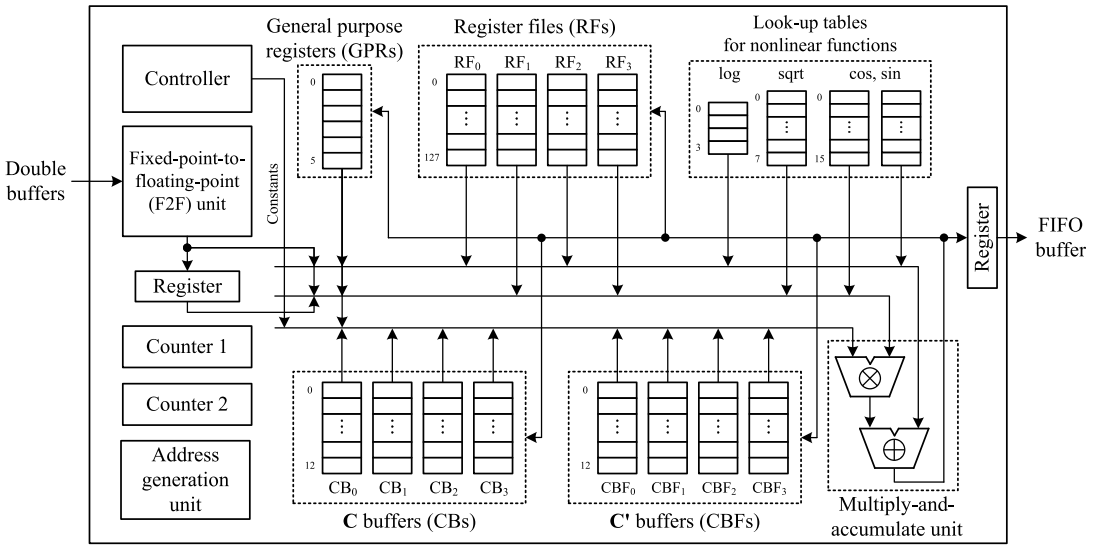

Fig. 6. Overall architecture of the proposed MFCC extraction.

calculated as

$$Y[l] = \sum_{k=k'_{ll}}^{k'_{lu}} |H[k]| \cdot W'_l[k] \tag{11}$$

where $l = 0, 1, \ldots, L-1$ and $W'_l[k]$ is not overlapped because k'_{lu} is equal to $k'_{(l+1)l}$. As a result, the power spectrum is simply achieved by adding two common subexpressions as follows:

$$X[l] = \log(Y[l-1] + Y[l]). \tag{12}$$

Due to the sharing of common subexpressions, the number of operations and the number of memory accesses in mel filtering are reduced approximately by half compared with those of previous architectures in [7]–[9].

B. Floating-Point System and Bit-Width Optimization

Since many operations used in the MFCC algorithm depend on complex functions, such as square and logarithmic functions, their outputs are associated with a large dynamic range. Compared with the fixed-point representation, the floating-point representation can cover such a large dynamic range with a much smaller number of bits. In addition, the operation bit-width can be reduced further, grounded on the property that the resulting feature vectors are influenced by the order of magnitude of interim values. For these reasons, a floating-point representation is employed in this brief to implement the modified MFCC extraction algorithm described above.

The floating-point number system has a value of

$$(-1)^S \times F \times 2^E \tag{13}$$

where S, E, and F denote the sign, exponent, and fraction parts, respectively. The bit-width of E is set to a constant value larger than or equal to the upper bound in order to prevent any possible overflows and underflows, but the bit-width of F is determined by conducting two optimization processes to be explained next.

To optimize the size of a LUT, a gradual approach summarized in Fig. 4 is used to decrease the bit-width gradually. The input bit-width or output bit-width of a certain LUT is repeatedly decremented by 1 if the difference between the recognition rates achieved with the bit-width and that with double-precision floating-point operations is less than a specific threshold. Otherwise, we stop the process and start a new optimization process for another LUT.

To optimize the bit-width of an operation, we simulate the modified extraction algorithm by setting the bit-width of the fraction part to F for all arithmetic operations. During the simulation, the LUTs optimized by the above-mentioned procedure are used to estimate the overall effect. Intensive simulations have been carried out to optimize the bit-widths of F while maintaining recognition rate high.

IV. Proposed Architecture for MFCC Extraction

This section presents a new floating-point MFCC extraction architecture derived to realize the MFCC extraction with a small hardware cost. This approach is completely different from [7]–[9] that have utilized a separate hardware unit for each operation. The proposed architecture is described with setting N to 256, M to 13, and L to 32. For sound signals sampled with 16 bits at 16 kHz, in addition, the bit-widths of F and E in the floating-point representation are determined to 6 and 7 bits, respectively.

A speech recognition system is conceptually shown in Fig. 5. One of the buffers stores a half of a sound frame and the other buffer is used to save the remaining data of the frame. Since subsequent frames share a half frame, only one buffer is updated for the next sound frame. The MFCC extraction system can generate MFCC feature vectors continuously by alternatively accessing the double buffers.

By analyzing the dataflow of the modified MFCC algorithm, we propose a new MFCC extraction system implementable with a small hardware cost. The overall architecture of the proposed system is shown in Fig. 6, which consists of a multiply-and-accumulate (MAC) unit, an address generation unit, a controller, memories, and counters.

Though the proposed architecture has one MAC unit, it is sufficient to process the entire MFCC extraction in real time. The constraint for real-time processing is that the MFCC vectors of a frame should be computed in a time limit corresponding to a half frame. Accordingly, a frame should be processed in 8 ms. The total number of MAC operations in the modified MFCC extraction is ∼15k, meaning that the modified MFCC algorithm can be processed in real time if

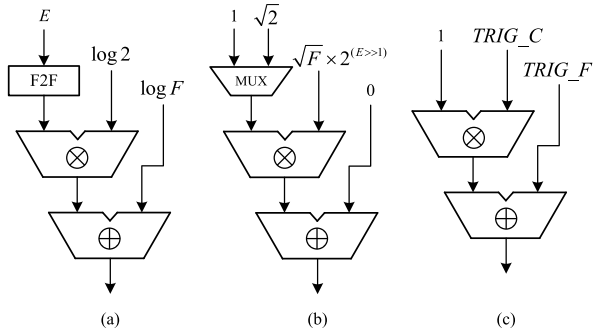

Fig. 7. Structures for computing nonlinear functions. (a) Logarithmic functions. (b) Square-root functions. (c) Trigonometric functions.

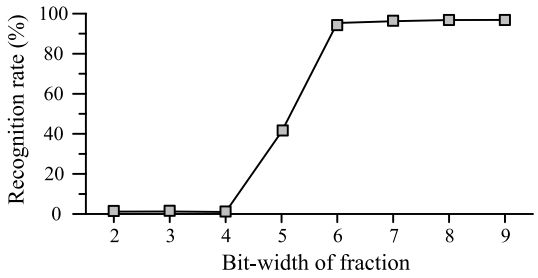

Fig. 8. Recognition rate versus the bit-width of the fraction part of the floating-point representation employed in MFCC extraction.

TABLE I
ENERGY CONSUMPTION OF THE PROPOSED MFCC EXTRACTOR

Technology	130 nm, 1.2 V
Operating frequency	50 MHz
Energy consumption	55 nJ/frame

~2-M MAC operations are supported in 1 s. This constraint is not hard for a modern embedded system to meet.

We now explain major blocks in detail. For floating-point operations, the fixed-point-to-floating-point unit is included to convert the sound data loaded from the double buffers to the floating-point representation. The MAC unit processes floating-point multiplication and accumulation in serial. Each operator consists of small fixed-point adders and multipliers, and the resultant fraction F is normalized to ensure that $1 \leq F < 2$ if $F \neq 0$.

The results of the MAC unit are saved into one of four memories: 1) general purpose registers (GPRs); 2) register files (RFs); 3) **C** buffers (CBs); and 4) **C**′ buffers (CBFs). The GPRs are used to store intermediate values such as the interim sound energy of a frame. The RFs are included to effectively compute such processes storing many values as FFT, mel filtering, DCT, and derivative computations. Grounded on the dataflow analysis of the modified MFCC algorithm, an efficient memory structure consisting of four separate RFs is derived. In terms of memory size, the proposed memory structure is more efficient than those of previous works, since it is shared with several processes.

To access an entity of a memory, the corresponding address is computed by employing counters. To fetch data for a MAC operation, each counter is increased by a certain amount. The proposed architecture utilizes two counters to generate two addresses needed to access two memories simultaneously.

The controller manipulates all the blocks to process the MFCC extraction algorithm, and its main role is to decide the input signals to be fed to the MAC unit and the storage to be used to store the results.

The nonlinear functions in the MFCC extraction are implemented as shown in Fig. 7. To calculate logarithmic and square-root functions, they are transformed into

$$\log(F \times 2^E) = E \times \log(2) + \log(F) \tag{14}$$

$$\sqrt{F \times 2^E} = \begin{cases} 1 \times (\sqrt{F}) \times 2^{(E \gg 1)}, & E \text{ is even} \\ \sqrt{2} \times (\sqrt{F}) \times 2^{(E \gg 1)}, & E \text{ is odd} \end{cases} \tag{15}$$

where $E \gg 1$ means that E is shifted right by 1. For (14) and (15), $\log(F)$ and $\sqrt{F}$ are stored in LUTs. Through the LUT optimization, three MSBs of F are turned out to be enough to compute approximate $\log(F)$. As the MSB of F is always 1 in computing $\log(F)$, the number of entries of the $\log(F)$ LUT is reduced to 4. In addition, three least significant bits (LSBs) of the fraction of $\log(F)$ have negligible effects on the recognition rate, so that $\log(F)$ is represented with 11 bits. For the square-root operation, four MSBs of F are considered to compute $\sqrt{F}$, and four LSBs of $\sqrt{F}$ are ignored. Since $\sqrt{F}$ is always positive, the $\sqrt{F}$ LUT consists of eight 9-bit entries.

To reduce the size of LUTs needed for trigonometric functions, the approximation in [12] is employed as follows:

$$\begin{aligned} \sin\left(\frac{\pi}{2} \cdot \frac{\text{addr}}{64}\right) &= \sin\left(\frac{\pi}{2} \cdot \frac{\alpha+\beta+\gamma}{64}\right) \\ &\simeq \underbrace{\sin\left(\frac{\pi}{2} \cdot \frac{\alpha+\beta}{64}\right)}_{TRIG\_C} + \underbrace{\cos\left(\frac{\pi}{2} \cdot \frac{\alpha}{64}\right) \sin\left(\frac{\pi}{2} \cdot \frac{\gamma}{64}\right)}_{TRIG\_F} \end{aligned} \tag{16}$$

where $0 \leq \text{addr} < 64$, $\alpha = \text{addr}$ AND $110000_{(2)}$, $\beta = \text{addr}$ AND $001100_{(2)}$, and $\gamma = \text{addr}$ AND $000011_{(2)}$. The first term *TRIG_C* is coarse grained and the second term *TRIG_F* is fine grained in (16), and they are stored in separate LUTs.

V. EXPERIMENTAL RESULTS

The modified MFCC algorithm is simulated to investigate how the recognition rate varies according to the bit-width of F. For the classification method, a hidden Markov model is used. Experiments are conducted for 1-s utterances obtained from 70 mixed-gender speakers. Two sets of 452-word samples are collected from each speaker. The simulation results are shown in Fig. 8. The recognition accuracy obtained from the exact MFCC algorithm is 98.5%. When the bit-width is reduced to 6 bits, the recognition rate is a little degraded to 96.1%. Therefore, the operation bit-width is set to 14 bits, including a 7-bit exponent and a single-bit sign, which is significantly reduced compared with that in [9]. For FFT magnitude computation, especially, the bit-width is reduced by a factor of 3.5.

To measure energy efficiency, the proposed floating-point MFCC extractor is synthesized in a 130-nm CMOS process. Table I summarizes the energy consumption per frame. Furthermore, the design is also implemented on a Xilinx field-programmable gate array (FPGA) to compare it with the previous works, as most of the designs in [7]–[9] were realized on FPGAs. The comparison results are summarized in Table II. Compared with the previous architecture, the proposed extractor is realized with a relatively small area, particularly in LUTs, because of the bit-width optimization and LUT unification. Moreover, the numbers of logic slices and multipliers are small in the proposed extractor, as a single MAC unit is used for entire computations. The equivalent gate counts of a four-input LUT, a flip flop, and a multiplier are estimated as 10, 5, and 2000, respectively. Since the energy consumption

TABLE II
COMPARISON OF MFCC EXTRACTION SYSTEMS

Architecture	This paper	[7]	[8]	[9]
FPGA	XC4VLX15	Spartan-3A DSP 1800	XC2V6000	XC3S200
Sound bit-width	16 bits	16 bits	16 bits	12 bits
Sound sampling rate	16 kHz	16 kHz	16 kHz	8 kHz
Samples/frame	256	512	256	200
Mel filters	32	24	-	26
Feature number	39	39	34	26
Accuracy	96.1 %	59.9 %	69.6 %	-
Logic slices	1397	2172	8696	2983
4-input LUTs	739	3010	16317	4218
Flip flops	2485	3106	9187	3974
Multipliers	1	8	1	15
Latency (cycles/frame)	14601	131200	54835	14272
Total equivalent gate count	21815	61630	211105	92050
Normalized energy consumption	1	25.4	36.3	4.1

depends on both hardware complexity and operating time, the product of them is used as a relative measure of energy consumption. Though the latency taken in the proposed extractor is slightly longer than that in [9], the normalized energy consumption is still much smaller due to the significantly reduced hardware complexity.

VI. CONCLUSION

An energy-efficient MFCC extraction architecture has been presented for speech recognition. The MFCC extraction algorithm is modified to minimize computation time without degrading the recognition accuracy noticeably. In addition, the proposed architecture employs floating-point arithmetic operations to minimize the operation bit-width and the total size of LUTs, while [7]–[9] have relied on fixed-point operators. Furthermore, a floating-point MAC unit and memories are shared with many processes to reduce hardware complexity and energy consumption remarkably. The effectiveness of energy consumption makes the proposed architecture a promising solution for energy-limited speech recognition systems.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generat. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[2] H.-W. Hon, "A survey of hardware architectures designed for speech recognition," Dept. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CS-91-169, Aug. 1991.

[3] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993, pp. 1–9.

[4] D. R. Reddy, "Speech recognition by machine: A review," *Proc. IEEE*, vol. 64, no. 4, pp. 501–531, Apr. 1976.

[5] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 28, no. 4, pp. 357–366, Aug. 1980.

[6] S. Nedevschi, R. K. Patra, and E. A. Brewer, "Hardware speech recognition for user interfaces in low cost, low power devices," in *Proc. 42nd DAC*, Jun. 2005, pp. 684–689.

[7] N.-V. Vu, J. Whittington, H. Ye, and J. Devlin, "Implementation of the MFCC front-end for low-cost speech recognition systems," in *Proc. ISCAS*, May/Jun. 2010, pp. 2334–2337.

[8] P. EhKan, T. Allen, and S. F. Quigley, "FPGA implementation for GMM-based speaker identification," *Int. J. Reconfig. Comput.*, vol. 2011, no. 3, pp. 1–8, Jan. 2011, Art. ID 420369.

[9] R. Ramos-Lara, M. López-García, E. Cantó-Navarro, and L. Puente-Rodriguez, "Real-time speaker verification system implemented on reconfigurable hardware," *J. Signal Process. Syst.*, vol. 71, no. 2, pp. 89–103, May 2013.

[10] D. G. Childers, D. P. Skinner, and R. C. Kemerait, "The cepstrum: A guide to processing," *Proc. IEEE*, vol. 65, no. 10, pp. 1428–1443, Oct. 1977.

[11] W. Han, C.-F. Chan, C.-S. Choy, and K.-P. Pun, "An efficient MFCC extraction method in speech recognition," in *Proc. IEEE ISCAS*, May 2006, pp. 145–148.

[12] D. A. Sunderland, R. A. Strauch, S. S. Wharfield, H. T. Peterson, and C. R. Cole, "CMOS/SOS frequency synthesizer LSI circuit for spread spectrum communications," *IEEE J. Solid-State Circuits*, vol. 19, no. 4, pp. 497–506, Aug. 1984.