INTEGRATION, the VLSI journal 76 (2021) 69–75

Contents lists available at ScienceDirect

Integration

journal homepage: www.elsevier.com/locate/vlsi

A low latency modular-level deeply integrated MFCC feature extraction architecture for speech recognition

Bibin Sam Paul S *, Antony Xavier Glittas, Lakshminarayanan Gopalakrishnan

Department of Electronics and Communication Engineering, National Institute of Technology Tiruchirappalli, Tamil Nadu, India

ARTICLE INFO

Keywords:
Automatic speech recognition
MFCC
ASIC
FPGA
Fast fourier transform

ABSTRACT

In this paper, a low-complex chip to extract the Mel Frequency Cepstral Coefficient for a speech recognition system is presented. The architecture can operate in a continuous-flow manner to process streaming or the stored speech signal at high speed. The frame-overlap Hamming window, DFT and Mel-filter bank computations are deeply integrated to share memory buffers and avoid bit-reversal circuit to reduce area and latency. Moreover, normalised energy consumption and area delay product are reduced by 32%, and speed is increased by 5.2 times compared to prior works. Further, the fixed-point word-length is optimised to minimise the area without affecting the accuracy.

1. Introduction

In the last few decades, substantial research work has been carried out in Automatic Speech Recognition (ASR) area, and significant technological progress made in this field. The speech contains the organization of a set of small elementary sounds called phenome and syllables that form a word, and the recognition task is to find the word from the collection of the observation. The significant parameters in the speech are extracted using a variety of dimensional reduction algorithms like Linear Predictive Coding (LPC) coefficients, Linear Prediction Cepstral Coefficients (LPCC) and Mel Frequency Cepstral Coefficients (MFCC). Among this MFCC feature is robust and immune to background noise compared to other algorithms [1,2]. The extracted observations are compared with the trained classifiers to find the best match for speech recognition. The popular classifier used in speech recognition are Hidden Markov Models (HMM) Vector Quantization (VQ), Gaussian Mixture Model (GMM) and Support Vector Machine (SVM). Speech recognition engines are generally equipped with a statistical classifier which has better performance compared to conventional pattern matching based on templates or spectral distance measure.

MFCC spectral feature is the most common choice for an automatic speech recognition system. MFCC algorithm, implemented on general purpose processors, DSPs, FPGAs, and ASIC, are reported in the literature [3–14]. VLSI architectures always have a trade-off between area speed and power by optimising the architecture either for area or processing speed. However, hardware implementation using FPGAs provide greater flexibility and reconfiguration capability to optimise a design suitable for the application requirements. Energy and hardware efficient implementation of MFCC extractor by modifying the algorithm without affecting the accuracy is proposed in Ref. [8,12]. MFCC algorithm implemented in FPGA targeted to minimise area by architecture modification is reported in Ref. [9–11]. MFCC feature extractor designed using standard IPs like Fast Fourier transform (FFT), CORDIC cores, as well as system generator tools, are also reported in the literature [16].

ASIC implementation targeted to minimise silicon area speed, and better accuracy are found in Ref. [5–7]. In Ref. [13] a dynamic ASIC based MFCC hardware architecture supporting re-configuration through AHB interface at the chip level is proposed. MFCC computation involves a series of complex mathematical operations computed step by step, and therefore it requires integration and optimisation of computation units to minimise silicon area. MFCC algorithm strength reduction techniques mostly focused on the Discrete Fourier transform (DFT) [5,8] and Mel filter bank computations [11,14].

Nowadays, the audio database has become massive, and modern applications demand high-speed computation to process them. Our proposed work is to design a low complex chip architecture to extract MFCC features that support continuous-flow operation to process streaming or stored speech signal at high speed as well as to occupy less silicon area. The area and latency are minimised by deeply integrating the computationally intensive modules such as frame-overlap Hamming window, FFT and Mel-filter bank computations. Frame-overlap Hamming window and DFT computation modules are integrated to share the

* Corresponding author.
E-mail addresses: nglbibin@gmail.com (B.S. Paul S), glittas@gmail.com (A.X. Glittas), laksh@nitt.edu (L. Gopalakrishnan).

https://doi.org/10.1016/j.vlsi.2020.09.002
Received 11 April 2019; Received in revised form 4 December 2019; Accepted 1 September 2020
Available online 16 September 2020

memory buffers, FFT and Mel-filter bank computation modules are integrated to eliminate bit-reversal circuit by modifying the Mel-filter bank architecture to process the spectrum samples independent of the input order.

The organization of the paper is as follows. Section II describes the overview of the MFCC algorithm and its mathematical formulation. Section III describes the detailed architecture of the entire system and the main characteristics of the proposed architectures. Section IV shows the experimental results and section V presents the conclusions.

2. MFCC feature extraction algorithm

MFCC algorithm extracts significant features from the speech signal by dividing the samples into discrete overlapping frames as the speech signal are stationary [2] over 10–30 ms. The mathematical model of the MFCC algorithm is given in Fig. 1, which comprise of complex mathematical operation like DFT, Logarithmic Mel-scale energy spectrum and cepstrum coefficient computation.

The speech signal $s(n)$ is pre-emphasized using equ. (1) to amplify the high frequency components in order to compensate for the attenuation effect caused during the physiological speech production process.

$$p(n) = s(n) - 0.97s(n-1) \tag{1}$$

The resulting signal is segmented into frames of N samples and then multiplied by the Hamming window using equ. (2) to smoothen the signal ends.

$$x(n) = p(n)\left\{0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)\right\} \tag{2}$$

The first component of the feature vector, logarithmic energy of the frame is computed using equ. (3)

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

The energy spectrum of the time domain signal $x(n)$ is computed using FFT as given by equ. (4,5).

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi n}{N}} \tag{4}$$

$$|X(k)|^2 = (\mathrm{Re}\{X(k)\})^2 + (\mathrm{Im}\{X(k)\})^2 \tag{5}$$

The logarithmic energy spectrum over the nonlinear Mel-scale for frame m corresponding to the indexed filter l is computed using equ. (6).

$$E_m(l) = \log\sum_{k=0}^{N-1} |X_m(k)|^2 H_l(k) \tag{6}$$

The $H_l(k)$ in equ. (6) define filter bank with M filters ($l = 1, 2, 3....M$), where the filter is triangular filter given in equ. (7)

$$H_l(k) = \begin{cases} 0 & k < f[l-1] \\ \dfrac{k - f[l-1]}{f[l] - f[l-1]} & f[l-1] \le k \le f[l] \\ \dfrac{f[l+1] - k}{f[l+1] - f[l]} & f[l] \le k \le f[l+1] \\ 0 & k > f[l+1] \end{cases} \tag{7}$$

The cepstrum coefficients are computed by inverse cosine transform as per equ. (8), and the resultant coefficients represent the features of the L − 1 band, where $n = 1, 2...L-1$ is the coefficient index.

$$C_m(n) = \sum_{l=1}^{M} E_m(l)\cos\left(n(l-0.5)\frac{\pi}{M}\right) \tag{8}$$

The dynamic first order and second order time derivative features are obtained using (9) and (10) respectively.

$$\Delta C_i = C_{i+2} + C_{i+1} - C_{i-1} - C_{i-2} \tag{9}$$

$$\Delta^2 C_i = \Delta C_{i+2} + \Delta C_{i+1} - \Delta C_{i-1} - \Delta C_{i-2} \tag{10}$$

In this work, the speech signal sampled at 16 KHz is segmented into frames of 16 ms with 8 ms (50%) overlap and the Hamming window is applied to smoothen the frame ends. The magnitude spectrum of each 256 sample speech frame is computed using FFT. For each frame, Mel-scale logarithmic energy spectrum centered around 32 triangular filter banks are computed over the frequency band with lower and upper bound as 130 Hz and 6800 Hz respectively [17]. Cepstrum coefficients are computed using 32 point Inverse FFT and correspondingly $\{C(0), C(1, 2.....12)\}$ coefficients are obtained for every frame, where $C(0)$ represents the energy of the frame. The first order and second order derivatives along with energy coefficients form a 39D coefficients for every frame.

3. Hardware architecture for MFCC

The architecture design of the MFCC extractor chip is organised into three sub-system levels as window and FFT sub-system, logarithmic Mel-scale energy sub-system and cepstrum and derivative sub-system as shown in Fig. 2. The sub-systems are organised based on the sharing of the hardware resources. The window and FFT sub-system integrate the hamming window and FFT module to share a common buffer for frame formation of the window module and for generating two parallel streams for the FFT module. The logarithmic Mel-scale energy computation subsystem reuses one logarithm module to process the 32 Mel-filter outputs for computing the logarithmic Mel-scale energy spectrum of a frame. The remaining smaller blocks are combined as cepstrum

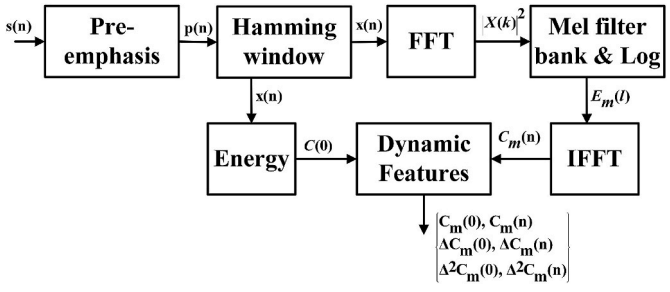

Fig. 1. Mathematical model of MFCC algorithm.

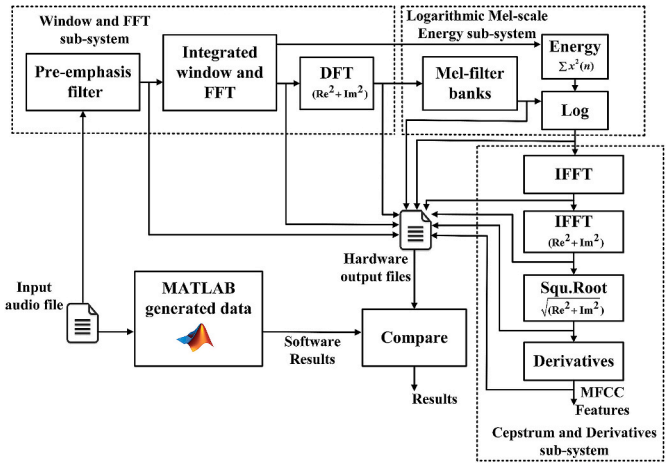

Fig. 2. MFCC functional block diagram and verification system.

& derivative sub-system for processing the cepstrum coefficients of the frame. The data format used for the implementation is optimised fixed point 2's complement representation considering the silicon area and accuracy requirement. The functional verification of the MFCC extractor system, as well as bit-width optimisation of every module, are carried out to compute the error between the software results and the hardware results with the help of suitable simulation models created using MATLAB.

3.1. Window and FFT sub-system

The pre-emphasis filter used to pre-process the speech signal is implemented without using a multiplier as proposed in Ref. [8]. The binary representation of the constant 0.97 in the pre-emphasis filter equ. (1) is '11111000'. As the number of 1's decide the number of adders the direct implementation of 0.97 using a constant coefficient multiplier result in 5 adders. In Ref. [8] the constant coefficient 0.97 is approximated as 31/32 so that the filter expression is modified to have a constant coefficient of 1/32. Therefore the circuit uses a delay element, a five-position shift register, adder, and a subtracter to replace the constant coefficient multiplier. The segmented sound frame formation with 50% overlap is shown in Fig. 3, with each frame segmented as lower (0 to N/2 − 1) and upper (N/2 to N − 1) half denoted as L and U respectively. Frames are processed as per the frame order from frame 1 to frame 'm' till the end of the speech stream. Frame 1 upper half is the frame 2 lower half and frame 3 lower half is frame 2 upper half for every pair of the frame considered with only change in the Hamming window multiplication coefficient. To divide the speech samples into discrete overlap frame without buffering the entire frame, and processing the frame paralley, an optimised architecture is proposed as shown in Fig. 4. The architecture integrates frame-overlap Hamming window, and DFT computation into a single structure so that the frame-overlap buffer is shared with the FFT module efficiently to process the speech stream in a continuous-flow manner.

The 50% overlap-frame is generated by converting the single stream speech signal into two parallel streams ($x(n)$ and $x(n+N/2)$) without any additional control signal using a RAM based shift register. The lower and upper half of the frame corresponding to the two streams are multiplied with the Hamming window coefficients stored in the ROMs (LROM, UROM) using two multipliers (M_L, M_U). The circuit begins operation after the N/2 size register gets full, and correspondingly the frame generation process started. During the frame formation, the same frame is loaded into either one of the FFT processor memory (memory 1 or memory 2) and parallely the consecutive next frame lower half is buffered into the shift register. The FFT processor starts computing the spectrum of the first frame after N clock cycle, and parallely the second frame is loaded into the memory.

The address for the window ROMs and FFT processor memories are generated for every clock cycle using a mod-N/2 counter. In the proposed architecture while a frame is loaded into the FFT memory the FFT processor process the previous frame loaded in the memory without any clock cycle stalls in a continuous-flow manner to minimise the computation cycles required for DFT computation. The process repeats until all

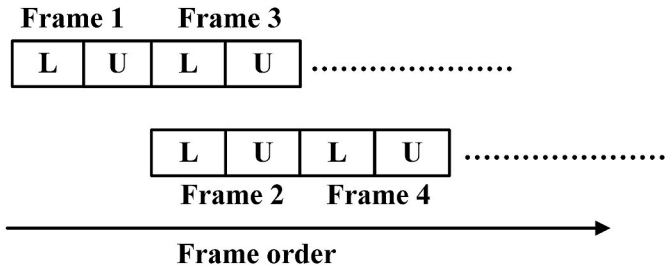

Fig. 3. Overlap frame formation.

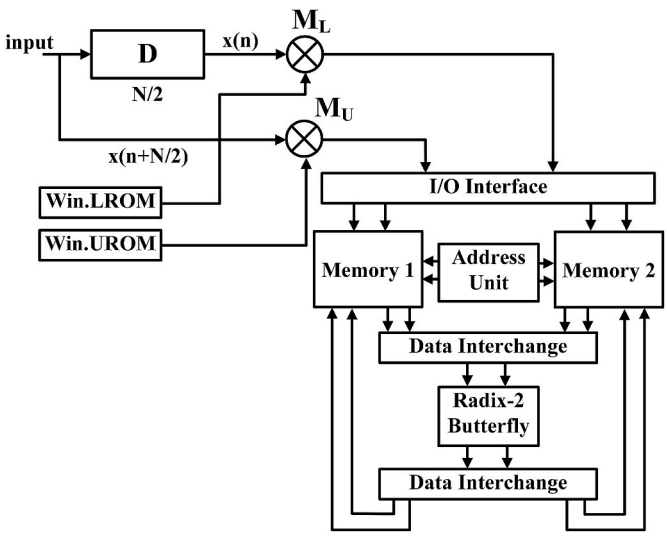

Fig. 4. Integrated frame-overlap FFT architecture.

the 'm' frames are processed. The proposed architecture reduces half of the memory location required for frame formation compared with the double buffers and dual clock auxiliary RAM architectures [9,12]. Furthermore, the design also avoids different clock domains as well as the MUX and separate control lines required to select between the double buffers. The proposed architecture can also be used to form a segmented audio frame with 75% overlap with the help of three N/4 size memory location forming four streams, and the radix-2 butterfly unit is replaced with radix-4 structure and correspondingly four multipliers, ROMs.

The FFT processor is designed using the memory based continuous-flow radix-2 in-place FFT architecture [15]. To maintain continuous-flow, the incoming data stream is swapped between memory 1 and memory 2 so that one memory is in computation mode and other is in I/O mode. The architecture comprises of two dual-port memory, an address generator unit which, generates the address to the two data and one twiddle factor memory, two data interchange unit, and a butterfly unit. The data interchange unit swaps between memory 1 and memory 2 to maintain continuous-flow of data. The in-place operation is ensured by giving proper control signals to memory, butterfly unit, and address unit to compute the stage-wise butterfly operation sequentially. The overall architecture uses two multipliers for windowing, one complex multiplier (three real multipliers) for twiddle factor multiplication.

After the initial latency of N clock cycle for every N/2 cycle computed DFT coefficients of the previous frame is available in either one of the memories alternatively. Due to two parallel continuous-flow operation, a pair of spectrum components are read out for every pair of sample loaded into the memory. The computed spectrum coefficients are in bit reversed order because of the Discrete in Frequency (DIF) FFT structure used for computation. The alternate terms in the bit-reversed output corresponding to the higher order bins (X (K + N/2)) are discarded for further processing. Hence the readout two parallel FFT output is downsampled, serialized into a single stream and correspondingly the clock domain change is avoided for the subsequent processing blocks. The squared energy spectrum is computed according to equ. (5) using a multiplier, shared between real and imaginary part via a MUX, adder, and a delay element arrangement.

Alternatively, for higher throughput applications, the same architecture can also be realised using two parallel multi-path delay commutator (MDC) FFT structure [19] integrated with the hamming window module as shown in Fig. 5. However, this architecture uses more area compared to the in-place architecture because of pipeline operation. The two parallel MDC architecture given in Fig. 5 uses DIF radix-2 butterfly unit (BF2) MDC switch (SW), delay elements (D) and a

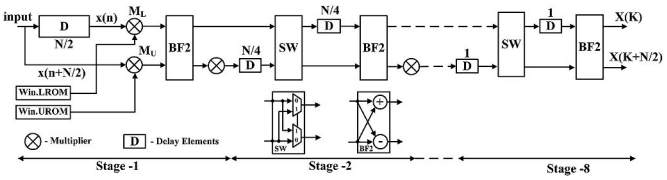

Fig. 5. Fully pipelined integrated frame-overlap FFT architecture.

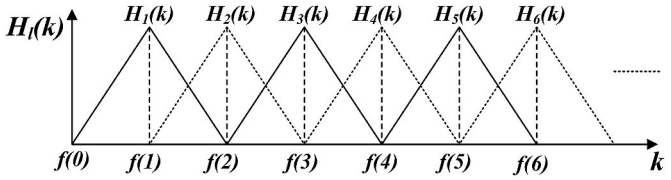

Fig. 7. Triangular Mel-filter banks.

twiddle multiplier considered as one processing element (PE). The MDC architecture uses 7 (log_2N-1) complex multipliers, 16 (2* (log_2N) adders and 382 (3 N/2 -2) registers. MDC structure produces two bit reversed outputs per cycle $X(K)$ and $X(K+N/2)$.

3.2. Logarithmic Mel-scale energy sub-system

The Mel-scale energy spectrum of every frequency band inside a frame is computed using the proposed architecture given in Fig. 6, and the resultant is taken log to compute the logarithmic Mel-scale energy spectrum according to equ. (6). The proposed architecture divides M filter banks into odd, and even group and each group is computed in parallel since the frequency spectrum centered on an even numbered filter bank say f_2 depends only on either f_1 or f_3 odd numbered filters as shown in Fig. 7. Therefore for every kth point spectrum, one pair (odd-even) of Mel-filter coefficient is non-zero and remaining are zero. The architecture is designed to share two multipliers (M_1, M_2) with M filter banks whereas compared to Ref. [14] our architecture uses 2:1 MUX in compared to the 3 input MUX used to select either from a zero or an active left and right band. The M filter outputs are serialized to make use of one log component to compute logarithmic Mel-scale energy spectrum computation.

The proposed architecture minimises the area by integrating the FFT and Mel-filter module to process spectrum components in bit-reversed order. To eliminate the bit-reverse circuit the filter coefficients of odd and even filters ($H_{2l-1}(k)$ *and* $H_{2l}(k)$ where$l = 1, 2 \ldots M/2$) are computed offline, and the non-zero coefficients are stored separately in odd and even ROMs. The control signals required to select an active odd-even filter bank pair are stored in a separate ROM with active filters as logic '1' and remaining as logic '0'. The bit reversed operation is ensured by giving the bit-reversed address to the control and coefficient ROMs thereby eliminating the re-ordering RAM as well the FFT and I/O mode switching schemes required in FFT processor [18] to process and read out data in normal order.

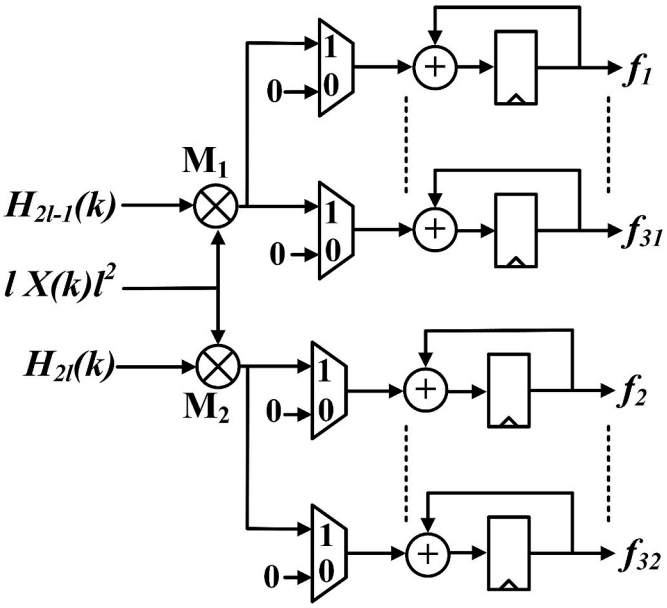

Fig. 6. Mel-scale energy spectrum computation architecture.

3.2.1. Fast binary logarithm module

The architecture for computing logarithm is implemented based on the fast binary logarithm algorithm [20]. The proposed 16-bit log computation architecture with a 4-bit exponent and 12-bit mantissa part is shown in Fig. 8. For the input x the exponent (2^y) is directly computed by detecting the position of leading one from MSB followed by a LUT to decode the corresponding powers of 2. LOD circuit is designed based on [21], which uses basic 2-input logic gates viz., AND, OR, and NOT.

The mantissa bits are computed one by one according to the required accuracy. The input is scaled ($2^{13} \le x \le 2^{12}$) first by a MUX based barrel shifter followed by squaring the scaled value and check $x^2 \ge 2^{13}$ using a comparator, if true then set mantissa bit to '1' and right shift x^2 (divide by 2) otherwise set mantissa bit to '0' and leave x^2 unchanged. Finally, the natural logarithm is computed using the constant multiplier architecture to minimise area. The main advantage of the proposed architecture are, mantissa bits are directly the comparator output without any logic computation and, the number of bits can be reconfigured based on the accuracy requirement. A MAC unit is attached with the Log unit using a MUX to compute the logarithmic frame energy according to (3).

3.3. Cepstrum and derivative sub-system

The cepstrum coefficients of the M logarithmic Mel-scaled energy spectrum coefficients are computed using inverse FFT (IFFT). IFFT is computed using FFT architecture with only the twiddle factor replaced by its complex conjugate, and the resultant is divided by N. The same in-place continuous-flow FFT architecture given Fig. 4 is used for IFFT computation with the two memories are replaced by a single memory to account for the dimensional reduction occurred in the input to the IFFT module. The architecture comprises of one dual-port memory, an address generator unit which, generates the address to the data and twiddle factor memory and a butterfly unit. The computed Mel-energy coefficients are available at the input of the IFFT module after 2N

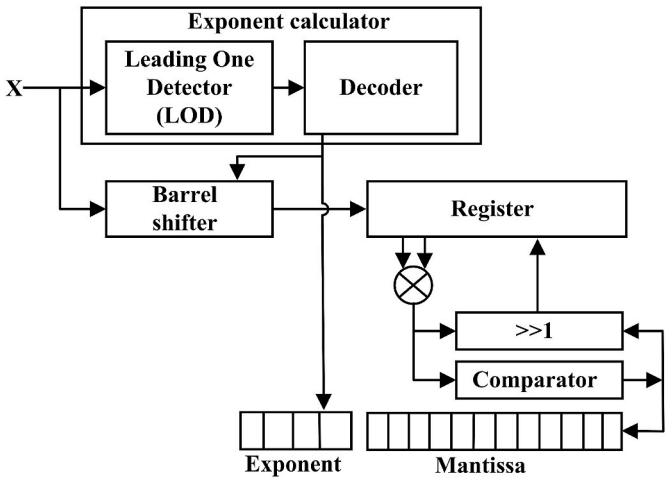

Fig. 8. Log computation Unit.

cycles. Subsequently, after the initial latency of N/2 cycle for every energy coefficient loaded into the memory the computed cepstrum coefficient of the previous frame is read out from the IFFT memory in a continuous-flow manner.

3.3.1. Reordering circuit

The IFFT processor uses the DIF FFT structure for computation, and correspondingly the readout coefficients are in bit-reversed order. The coefficients are reordered using the circuit [23] shown in Fig. 9 to select the 12 feature vectors. The circuit comprises of four 2:1 MUX and a 7D and 2D delay registers. The upper half coefficients 16 to 32 is directly eliminated without entering into the circuit by giving control signals to M1 and M2 for every two cycles.

The remaining 16 points (0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15) are reordered by first storing 0,8,4,12,2,10,6 data in the 7D shift register and the control signals for the M_1 and M_2 MUX are swapped for every clock result in (0, 1, 4, 5, 2, 3, 6, 7, 8, 9, 12, 13, 10, 11, 14, 15) at the output of M_2. Subsequently, after loading to the 2D register, the control signal for the M_3 and M_4 MUX are swapped for every two cycles to generate the normal order ($x_{no}(n)$) output. The first and second order derivatives of the cepstrum coefficients are computed according to equ. (9,10) using the combination of adders, subtractors and delay elements. The square root function is implemented based on [22] using shifters adders and subtractors.

4. Experimental results

The MFCC feature extraction system contains complex mathematical operations like FFT, Mel-filter bank computation and non-linear functions like square, logarithm and square root operations. The dynamic range of the inputs and outputs of each sub-module varies by a considerable margin which results in the loss of accuracy if a constant fixed-point representation is used. To minimise the loss of accuracy variable precision fixed-point representation is used in the sub-modules of the proposed architecture as given in Ref. [10]. In variable precision fixed-point representation, each module has a different data width depends on the dynamic range of the operation. The data width of the input and output depends on the number of bits selected for integer and fraction part, where the integer part minimise the overflow error and the fraction part decides the precision. In the proposed architecture the integer part of each module is selected by analysing the dynamic range of each module by giving a bounded input ±1 represented as (s1.14) data format with one bit for sign one bit for integer and 14 bit for the fraction. The fraction part is optimised by considering the trade-off between accuracy and hardware complexity.

The proposed MFCC extractor is optimised to reduce the resource utilization without significantly sacrificing the accuracy by varying the fraction bit width of the individual modules from 2 to 14. The effect of the increase in fraction bit-width on the relative error percentage between software floating point implementation and hardware fixed-point implementation of FFT, Mel-filter bank, Log, and IFFT modules are shown in Fig. 10. The other smaller modules are not plotted in the graph because only the integer part width is varied and the fraction part width is kept unchanged. In the FFT module, the error decreases with a sharp slope between 2 and 10, and then improvement becomes negligible, based on this fraction bit width is set to 10 bits corresponding to a total data width of 20 with a 1-bit sign, 9-bit integer part and 10-bit fraction

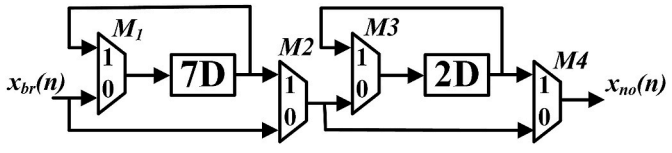

Fig. 9. Re-ordering circuit.

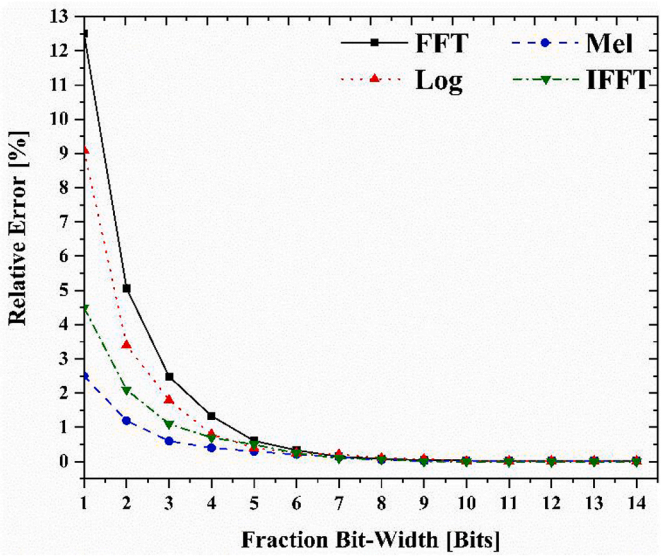

Fig. 10. Relative error percentage of MFCC modules.

part for the FFT implementation. Similarly, the same process is repeated for all other modules to fix the bit width. In Fig. 10, although the percentage of error is small for Log and IFFT modules with a smaller number of fraction bits, a higher number of bits are allocated for the fractional part. This is done to counter the error propagating from the previous stage. The overall integrated system is optimised by considering both the percentage error and word recognition accuracy. The optimised output data width and data format of all the sub-modules in the MFCC algorithm are given in Table 1. To evaluate the word recognition accuracy of the proposed MFCC extractor a Hidden Markov Model (HMM) based trained classifier is used. The classifier is trained using the dataset collected from 32 speakers (18 males and 14 females). The dataset contains 1 s recorded 50 isolated words from each speaker totaling to 1600 utterances. The word recognition accuracy of the proposed hardware implementation is 94.8% compared to the software implementation accuracy of 96.7%.

The hardware resource utilization of the proposed MFCC extractor is evaluated by implementing the design on Xilinx FPGA and compare it with previously reported work as summarised in Table 2. Compared to the other reported architecture the proposed architecture uses less number of slice registers. However, in terms of LUTs other than [3,12] our architecture uses less number of LUTs because the former architectures use a serial approach compared to our continuous-flow approach. Due to the two parallel continuous-flow design, the proposed architecture generates the stream of feature vectors for every 128 speech samples. The computational modules of the proposed architectures are operated w. r.t the data clock except for the FFT, IFFT processor and the log module and correspondingly the total computation cycles

Table 1
Word lengths of MFCC modules.

Function	Data width	Data format (Integer. Fraction)
Pre-emphasis	16	s1.14
Integrated Windowing & FFT	20	s9.10
FFT ($Re^2 + Im^2$)	28	18.10
Mel-filter Bank	18	16.2
Log	16	s5.10
IFFT	18	s5.12
IFFT ($Re^2 + Im^2$)	22	10.12
Squ.Root $\sqrt{(Re^2 + Im^2)}$	11	5.6
Derivatives	12	s5.6

Table 2
FPGA implementation performance comparison.

Architecture	[3]	[4]	[10]	[12]	Proposed work
FPGA	Spartan-3A DSP 1800	XC2V6000	XC3S200	XC4VLX15	XC4VLX15
Sound Bit-width (bits)	16	16	12	16	16
Sound sampling Rate (KHz)	16	16	8	16	16
Samples/frame	512	256	200	256	256
Mel-filters	24	–	26	32	32
Feature Number	39	34	26	39	39
Accuracy (%)	59.9	69.6	–	96.1	94.8
Slice Registers	3106	9187	3974	2485	1984
Slice LUT	3010	16317	4218	739	4092
Multipliers	8	1	15	1	14
Latency (Cycles/frame)	131200	54835	14272	14601	2740
Equivalent Gate Count	61630	211105	92050	21815	78840
Normalised Energy Consumption	37.43	53.58	6.08	1.47	1
Area Delay Product	80.85	115.75	13.13	3.18	2.16

required to generate the feature vector is 2740, which is 5.2 times lesser compared to other architectures. To prove the efficiency of the proposed architecture the equivalent gate count is estimated based on [12] considering the factor 5, 10, and 2000 respectively a slice register, a LUT, and a multiplier. The energy consumption of the architecture is also estimated as per [12], in which the relative measure of energy consumption is defined as the product of hardware complexity and operating time. In the case of normalised energy consumption, the energies consumptions of other reported works are normalised w. r.t the proposed architecture. The product of the equivalent gate count and the computation time for the operating frequency of 100 MHz is computed to obtain the area-delay product. The significant reduction in the computation cycles and the efficient area utilization of the proposed architecture result in a 32% reduction of area-delay product as well as normalised energy consumption compared to other architectures.

The proposed MFCC architecture is synthesized using CMOS 130 nm technology and the functional verification, logical equivalence and timing closure of the post layout netlist with parasitic extraction is performed to validate the performance of the MFCC chip. Fig. 11 shows the layout diagram of the chip comprising the core and IOs. The power consumption of the proposed MFCC architecture for the operating frequency of 100 MHz and supply voltage of 1.2 V is 1308 μW. From the power consumption, the energy consumption per frame is computed and compared with [12] to prove the energy efficiency of the proposed MFCC architecture as given in Table 3. Due to the reduction of computation time, the energy consumption per frame of the proposed architecture is significantly reduced compared to Ref. [12].

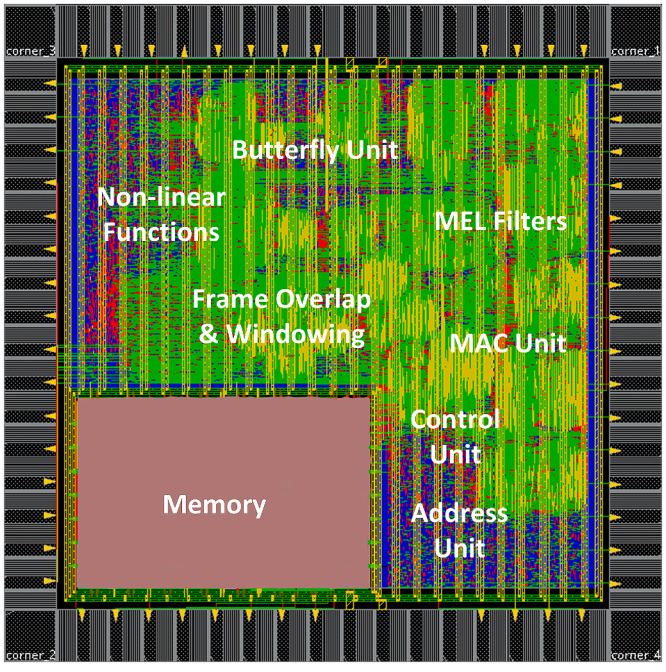

Fig. 11. Layout of the chip.

Table 4 shows the ASIC implementation performance comparison with other reported works. Although the die size prescribed by the foundry is 1.525 × 1.525 mm the core area of our IC is 0.9 × 0.9 $mm^{2,}$ which is less compared to other reported ASIC implementation [7,13]. The processing time of the proposed architecture is significantly less compared to other architectures, although the architecture in Ref. [13] is operated in high frequency the proposed design process the frame at a much higher speed.

5. Conclusion

This paper presented VLSI architecture of MFCC feature extraction chip. The proposed architecture process speech input in a continuous -flow manner to minimise the area and latency. The area and latency are minimised by integrating the computationally intensive frame-overlap Hamming window, DFT and Mel filter bank computation effectively to share memory buffers and avoid bit-reversal circuit. A novel logarithmic computation block is designed based on the fast binary logarithmic algorithm for efficient reconfiguration of the bit-width. Word length of individual modules is also optimised to achieve overall accuracy. These attributes show the effectiveness of the proposed architecture to improve the overall performance.

Authorship contributions

S. Bibin Sam Paul: Conception, Methodology, Formal analysis, Software, Drafting, editing the manuscript X. Antony Xavier Glittas: Conception, Methodology, Formal analysis, Software, Investigation, Validation, Drafting, editing the manuscript, G. Lakshminarayanan: Investigation, Validation, Reviewing the manuscript critically for important intellectual content, Supervision, Funding acquisition, Project administration.

Table 3
Energy consumption performance comparison.

Parameters	[12]	Proposed work
CMOS Technology (nm)	130	130
Operating Frequency (MHz)	50	100
Energy Consumption (nJ/frame)	55	35.84

Table 4
ASIC implementation performance comparison.

Parameters	[7]	[13]	Proposed work
CMOS Technology (μm)	0.6	0.13	0.13
Area (mm^2)	3.2 × 3.3	1.29 × 1.29	0.9 × 0.9
Operating Frequency (MHz)	50	500	100
Processing time (μs)	73.4	2348	27.4

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors thank Ministry of Information Technology (MeitY)-Government of India, DST-FIST for funding the lab facility for supporting this research under Grant number DST/ETI-324/2012 and SMDP-C2SD for providing EDA tool facilities for this research work.

References

[1] S. Davis, P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, IEEE Trans. Acoust. Speech Signal Process. 28 (4) (1980) 357–366.
[2] R. Vergin, D. O'Shaughnessy, A. Farhat, Generalized Mel frequency Cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition, IEEE Trans. Speech Audio Process. 7 (5) (1999) 525–532.
[3] N. Vu, J. Whittington, H. Ye, J. Devlin, Implementation of the MFCC front-end for low-cost speech recognition systems, in: Proceedings of IEEE International Symposium on Circuits and Systems, Paris, 2010, pp. 2334–2337.
[4] P. EhKan, T. Allen, S.F. Quigley, FPGA implementation for GMM-based speaker identification, Int. J. Reconfig. Comput. 2011 (3) (2011) 1–8.
[5] T. Kuan, J. Wang, T. Shang-Hung, Optimized Radix-2 FFT and Mel-Filter Bank in MFCC-Based Events Sound Recognition Chip Design for Active Smart Warming Care, International Conference on Orange Technologies, Xian, 2014, pp. 197–200.
[6] Jia-Ching Wang, Jhing-Fa Wang, Yu-Sheng Weng, Chip design of MFCC extraction for speech recognition, Integration 32 (1–2) (2002) 111–131.
[7] Jia-Ching Wang, Jhing-Fa Wang, Yu-Sheng Weng, Chip design of Mel frequency cepstral coefficients for speech recognition, in: IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings, Istanbul, Turkey vol. 6, 2000, pp. 3658–3661.
[8] Wei Han, Cheong-Fat Chan, Chiu-Sing Choy, Kong-Pang Pun, An efficient MFCC extraction method in speech recognition, in: IEEE International Symposium on Circuits and Systems, Island of Kos, 2006, pp. 146–148.
[9] V. Rodellar-Biarge, C. Gonzalez-Concejero, E. Martinez De Icaya, A. Alvarez-Marquina, P. Gómez-Vilda, Hardware reusable design of feature extraction for distributed speech recognition, in: AEE'07 Proceedings of the 6th Conference on Applications of Electrical Engineering, 2007, pp. 47–52.
[10] R. Ramos-Lara, M. López-García, E. Cantó-Navarro, et al., Real-time speaker verification system implemented on reconfigurable hardware, Journal of Signal Processing System (2013) 71–89.
[11] Gin-Der Wu, Ying Lei, Parallel Dual-Accumulator Based Mel Frequency Cepstral Coefficient for Speech Recognition, IET 4th International Conference on Intelligent Environments, Seattle, WA, 2008, pp. 1–4.
[12] J. Jo, H. Yoo, I. Park, Energy-efficient floating-point MFCC extraction architecture for speech recognition systems, IEEE Trans. Very Large Scale Integr. Syst. 24 (2) (2016) 754–758.
[13] T.C. Nguyen, L.D. Pham, H.M. Nguyen, B.G. Bui, D.T. Ngo, T. Hoang, A high performance dynamic ASIC-based audio signal feature extraction (MFCC), in: International Conference on Advanced Computing and Applications (ACOMP), Can Tho, 2016, pp. 113–120.
[14] M. Price, J. Glass, A.P. Chandrakasan, A 6 mW, 5,000-word real-time speech recognizer using WFST models, IEEE J. Solid State Circ. 50 (1) (2015) 102–112.
[15] P. Tsai, C. Lin, A generalized conflict-free memory addressing scheme for continuous-flow parallel-processing FFT processors with rescheduling, IEEE Trans. Very Large Scale Integr. Syst. 19 (12) (2011) 2290–2302.
[16] M. Bahoura, H. Ezzaidi, Hardware implementation of MFCC feature extraction for respiratory sounds analysis, in: 8th International Workshop on Systems, Signal Processing and Their Applications, WoSSPA), Algiers, 2013, pp. 226–229.
[17] S.K. Kopparapu, M. Laxminarayana, Choice of Mel filter bank in computing MFCC of a resampled speech, in: 10th International Conference on Information Science, Signal Processing and Their Applications (ISSPA 2010), Kuala Lumpur, 2010, pp. 121–124.
[18] R. Radhouane, P. Liu, C. Modlin, Minimizing the memory requirement for continuous flow FFT implementation: continuous flow mixed mode FFT (CFMM-FFT), in: IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century. Proceedings Geneva, Switzerland, vol. 1, 2000, pp. 116–119.
[19] S. He, M. Torkelson, Designing Pipeline FFT Processor for OFDM (de)Modulation, ISSSE, 1998, pp. 257–262.
[20] C.S. Turner, A fast binary logarithm algorithm [DSP tips & tricks], IEEE Signal Process. Mag. 27 (5) (2010) 124–140.
[21] K. Kunaraj, R. Seshasayanan, Leading one detectors and leading one position detectors - an evolutionary design methodology, Can. J. Electr. Comput. Eng. 36 (3) (2013) 103–110.
[22] Yamin Li, Wanming Chu, A new non-restoring square root algorithm and its VLSI implementations, in: Proceedings International Conference on Computer Design, VLSI in Computers and Processors, Austin, TX, USA, 1996, pp. 538–544.
[23] M. Garrido, J. Grajal, O. Gustafsson, Optimum circuits for bit reversal, IEEE Transactions on Circuits and Systems II: Express Briefs 58 (10) (2011) 657–661.