LOW POWER EMBEDDED SPEECH RECOGNITION SYSTEM BASED ON A MCU AND A COPROCESSOR

[1]*Peng Li,* [1]*Hua Tang,* [2]*Weiqian Liang**

[1]Dept. of Electrical and Computer Engineering, University of Minnesota, Duluth, MN, 55812, USA
Email: {lixxx988, htang}@umn.edu
[2]Dept. of Electronic Engineering, Tsinghua University, Beijing, 10084, China
Email: lwq@tsinghua.edu.cn

ABSTRACT

In speech recognition systems, CHMM (Continuous Hidden Markov Model) based speech recognition algorithms have the best accuracy but with the most computational cost. Neither General Purpose Processor (GPP) nor dedicated hardware implementation is a good solution for the algorithm, due to high power consumption for the former and lack of flexibility for the later. To reduce power consumption and enhance flexibility, this paper presents a speech recognition system composed of a coprocessor and a MCU. The coprocessor is a dedicated hardware design for Output Probability Calculation (OPC), which is the most computation-intensive part in CHMM, and MCU is a 32bit RISC (ARM). Tested with a 358-state 3-mixture 27-feature 800-word HMM, MCU operates at 40MHz and coprocessor operates at 10MHz to meet real-time requirement. The power consumption of MCU is 10mW, and coprocessor 1.8mW.

Index Terms— HMM, Speech recognition, FPGA, Coprocessors.

1. INTRODUCTION AND PREVIOUS WORK

CHMM based speech recognition algorithms have a high recognition accuracy for word recognition tasks [1] [2]. It is increasingly popular in mobile and other embedded applications. Most commercial products take GPP [1] [2] [3] as the main implementation approach for CHMM based embedded speech recognition systems. However, since the hardware is not a dedicated design for the algorithm, it is not able to efficiently process the vast amount of vector operations. Therefore GPP always requires a very high operation frequency to meet real-time requirement. As a result, the power consumption and cost of these systems are usually very high.

Dedicated hardware implementation of CHMM based speech recognition algorithms is an effective solution for low-power embedded system compared to GPP based implementation. For example, a scalable architecture implementing the whole speech recognition algorithm is reported in [4]. Its processing time with 800-word vocabulary is rather small (56.9μs/word). Its power consumption is 421.5mW. However, this architecture is only suitable for HMM with single Gaussian mixture distribution. For multiple Gaussian mixtures, hardware architecture needs to be adjusted, which reduces its flexibility. An HMM-based speech recognition Integrated Circuit in [5] could operate at 20MHz to finish recognition task. However, the word library has only 50 words, and it handles only double Gaussian mixtures. Besides, the look-up table based approach in [5] for OPC requires more memory resources.

To gain advantages from both, some work combines GPP and dedicated hardware implementation. In that case, dedicated hardware is designed for part of the speech recognition algorithm, leaving other operations to GPP [6] [7] [8]. In [6], FPGA implementation of Viterbi algorithm had been proposed, leaving other operations to Motorola 56002. However, the number of states is limited to 6 to meet real-time requirement. A coprocessor for Mahalanobis distance calculation had been designed in [7], leaving other operations to ARM7. Hardware implementation for OPC is proposed in [8]. However, the sequential approach to calculate Mahalanobis distance and add-log in [8] lowers the processing speed compared to the parallel approach.

In this paper, we propose a new speech recognition system by combining MCU and coprocessor. The coprocessor is a dedicated hardware design for OPC. MCU processes other operations needed by the algorithm and system control tasks. We target speech recognition systems with less demanding requirements on response time, such as those in telephone, language study machine and toy, where real time factor no more than 1 is acceptable [3]. The proposed system has the following characteristics. 1) The interface between coprocessor and MCU is standard SRAM interface, which makes the coprocessor easily controlled by MCU. 2) The overall system is easily reconfigurable through MCU when parameters of HMM change so that the system is very adaptable to different recognition tasks. 3) The coprocessor can calculate Mahalanobis distance and add-log in parallel. 4) Add-log calculation is based on polynomial fitting method [9] which is more area-saving compared to look-up table method. 5)

*This work is supported by Department of Electronic Engineering, Tsinghua University, Beijing, China (Contact person: Weiqian Liang, e-mail: lwq@tsinghua.edu.cn).

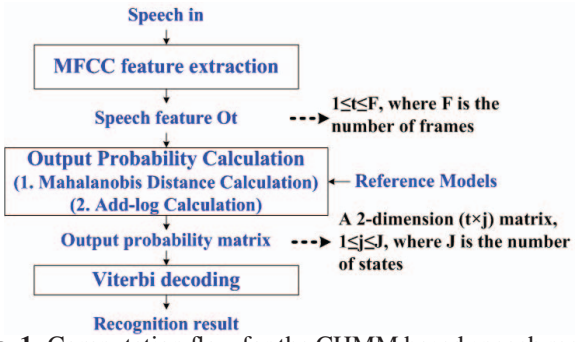

Fig. 1. Computation flow for the CHMM based speech recognition algorithm.

Single port SRAMs are adopted in the coprocessor to save hardware resource significantly. We implemented the whole speech recognition system by Samsung S3C44b0X [10] and Xilinx FPGA Virtex II XCV2000 [11].

The rest of the paper is organized as follows. Section 2 describes OPC in CHMM based speech recognition algorithms. Section 3 presents the implementation of the system. Section 4 gives the experimental results. Finally, conclusions are drawn in Section 5.

2. OUTPUT PROBABILITY CALCULATION

In CHMM based speech recognition algorithms, there are mainly three steps, MFCC (Mel-Frequency Cepstral Coefficients) feature extraction, OPC and Viterbi decoding [3] [7] [12]. The second step, OPC, is composed of two sub-steps, Mahalanobis distance calculation and add-log calculation. The computation flow of this algorithm is shown in Fig. 1.

In the following, let's define *F* is the number of frames, *J* the number of states, *G* the number of Gaussian mixtures, *M* the number of features, and *W* the number of words. Then, the processing time of MFCC feature extraction is T_{MFCC} = $F \times M \times P_{MFCC} / f_c$, Mahalanobis distance calculation is T_{MDC} = $F \times J \times M \times G \times P_{MDC} / f_c$, add-log calculation is T_{ALC} = $F \times J \times G \times P_{ALC} / f_c$, and Viterbi decoding is T_{VD} = $W \times P_{VD} / f_c$ [3] [7], where f_c is the operation frequency, P_{MFCC}, P_{MDC}, P_{ALC} and P_{VD} is the number of clock periods for the corresponding computation respectively. (The value of f_c, P_{MFCC}, P_{MDC}, P_{ALC} and P_{VD} is different for different implementations.)

We have evaluated the computation load for each step of this algorithm based on a 358-state 3-mixture 27-feature 800-word HMM in different implementations, such as a general-purpose 16bit fix-point DSP Uni-Lite [3] [13], a 32bit RISC MCU S3C44b0X [10], and an AMD Sempron 2800+ PC. The results in different implementations are shown in Table 1. It can be seen that OPC is the most computation-intensive processing step of the algorithm. However, OPC has the characteristic of regular data flow which will be illustrated below.

The output probability density function $\widetilde{b_j}(o_t)$ in logarithmic domain (to avoid underflow) is shown in (1) [3] [7] [12]. where o_t is the speech feature vector at frame *t*, c_{jg}, μ_{jg} and

Table 1. Distribution of computation load for each step of speech recognition algorithm in three implementations

	Uni-Lite	S3C44b0X	PC
MFCC feature extraction	**10.5%**	**8.7%**	**8.2%**
Output probability calculation	**70.9%**	**75.9%**	**81.2%**
(Mahalanobis distance calculation)	(55.7%)	(61.2%)	(64.7%)
(Add-log calculation)	(15.2%)	(14.7%)	(16.5%)
Viterbi Decoding	**18.6%**	**15.4%**	**10.6%**

Σ_{jg} is the weight, the mean and the covariance matrix respectively for state *j* and the g^{th} Gaussian mixture distribution.

$$\widetilde{b_j}(o_t) = log\sum_{g=1}^{G} c_{jg} b_j(o_t) = log\sum_{g=1}^{G} \{exp[-\frac{1}{2}(o_t-\mu_{jg})'\Sigma_{jg}^{-1}(o_t-\mu_{jg}) + log\frac{c_{jg}}{\sqrt{(2\pi)^M|\Sigma_{jg}|}}]\} \quad (1)$$

The first part in (1) is Mahalanobis distance calculation, which is shown in (2). The second part in (1) is add-log calculation, which is shown in (3).

$$-\frac{1}{2}(o_t-\mu_{jg})'\Sigma_{jg}^{-1}(o_t-\mu_{jg}) = -\frac{1}{2}\sum_{i=1}^{M}[(o_{ti}-\mu_{jgi})\delta_{jgi}]^2 \quad (2)$$

$$log\{\sum_{g=1}^{G} exp[Q_g]\} \quad (3)$$

where $Q_g = -\frac{1}{2}\sum_{i=1}^{M}[(o_{ti}-\mu_{jgi})\delta_{jgi}]^2 + d_{jg}$, $d_{jg} = log\frac{c_{jg}}{\sqrt{(2\pi)^M|\Sigma_{jg}|}}$, and δ_{jgi} is the square root of the i^{th} diagonal element in matrix Σ_{jg}^{-1} [4] [7]. It can be seen that Mahalanobis distance calculation in (2) could be converted to a group of multiplication and addition operations.

For add-log calculation in (3), let's define *F(a, b)* as follows:

$$F(a,b) = log[exp(a)+exp(b)]$$
$$max(a,b) + log[1+exp(-|a-b|)] \quad (4)$$

From (4), further define *G(x)* as follows [11]:

$$G(x=|a-b|) = log[1+exp(-x)] =$$
$$A_0 + xA_1 + x^2A_2 + ... + x^nA_n =$$
$$A_0 + x(A_1 + x(A_2 + ...(A_{n-1} + xA_n))) \quad (5)$$

with (4) and (5), we rewrite the add-log calculation in (3) as follows:

$$log\{\sum_{g=1}^{G} exp[Q_g]\} = F(F(F(Q_1,Q_2),Q_3),...,Q_G) \quad (6)$$

As a result, the add-log calculation in (3) could be converted to a group of basic algebraic calculations [9], where A_i is the coefficient of the polynomial (i=0,1,2,...,*n*, where *n* is the maximum power of *x* in (5)).

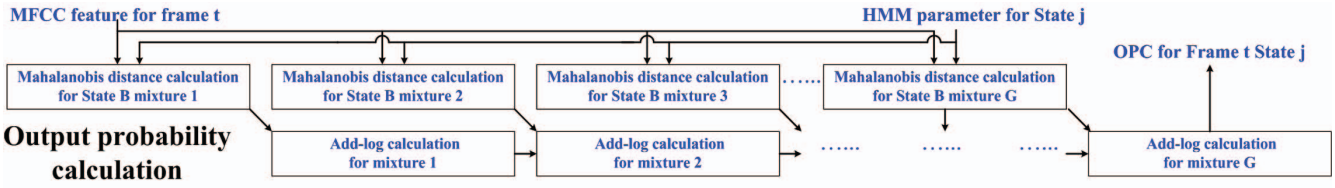

Fig. 2. Computation flow for OPC.

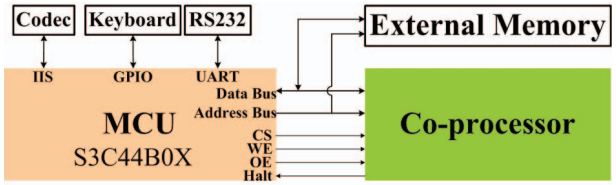

Fig. 3. Block diagram of the speech recognition system.

3. SYSTEM IMPLEMENTATION

3.1. Overall System Architecture

In Section 2, it can be seen that OPC is very suitable to dedicated hardware design. Moreover, add-log and Mahalanobis distance can be calculated in parallel. The key point that justifies this parallel processing is that the current add-log calculation is only related to the last Mahalanobis distance calculation with the same Gaussian mixture, which is shown in Fig. 2. Therefore we can compute Mahalanobis distance for mixture *g* and the add-log for mixture *g-1* at the same time.

We design a speech recognition system composed of a MCU and a coprocessor to achieve the best tradeoff between GPP and dedicated hardware design. The MCU will process MFCC feature extraction, Viterbi decoding, and system control tasks, and the coprocessor takes care of OPC. We made this decision because computation loads of MFCC feature extraction and Viterbi Decoding is small and dedicated hardware implementation for the entire speech recognition system would lack flexibility.

Fig. 3 shows the block diagram of the entire system. The interface between MCU and coprocessor is the standard SRAM interface except a "*Halt*" signal, which is used for interrupt. This interface makes the coprocessor easily controlled by various MCU, such as ARM [14] and MIPS [15].

3.2. Coprocessor Design

The architecture of the coprocessor is shown in Fig. 4. The block diagrams of *MDC (Mahalanobis Distance Calculation) Unit* and *ALC (Add-Log Calculation) Unit* are shown in Fig. 5. *MDC Unit* performs a four-stage pipeline operation for Mahalanobis distance calculation in (2). The total processing time is $(M + 3) / f_c$. *ALC Unit* is used to calculate polynomial addition in (5). The total processing time is $(n + 1) / f_c$. Usually, n is set to 6 to ensure high precision.

Fig. 6 shows the block diagram of *Interface Unit*. It is used to perform *max* and sum functions in (4), and iterations in (6). The processing time of OPC for one frame and one

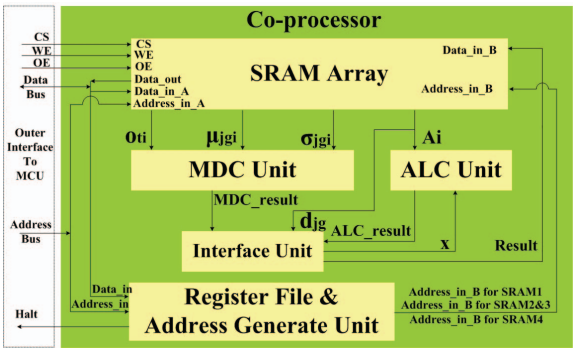

Fig. 4. Block diagram of coprocessor.

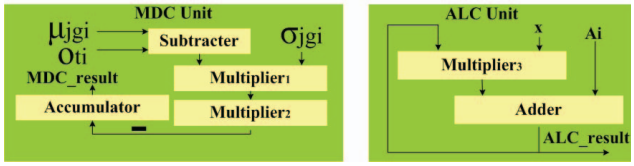

Fig. 5. Block diagram of MDC and ALC Unit.

state is $T_{OPC} = (G \times (M + 3) + (n + 1)) / f_c$, and the total processing time of OPC for all frames and all states is $F \times J \times T_{OPC}$, compared to the sum of T_{MDC} and T_{ALC} in Section 2, the processing time of OPC is reduced significantly.

Fig. 7 shows the block diagram of *SRAM Array*. There are two *Address_in* ports for the four SRAM blocks, and two *Data_in* ports for *SRAM4*. Although dual-port SRAMs may seem necessary, we can use single-port SRAMs with multiplexers controlled by both MCU and coprocessor since they do not need to access these SRAMs at the same time. This also helps reduce the hardware resources [7].

4. EXPERIMENT AND MEASUREMENT RESULTS

We used Xilinx Virtex II FPGA XCV2000 to implement the coprocessor. The design summary of Xilinx ISE shows that the total equivalent 2-NAND gate count for the design is 26K. Samsung S3C44b0x (containing an ARM core) [10] is used as MCU. Fig. 8 shows the complete speech recognition system.

Test for the system is based on the same HMM used in Section 2. Result shows that total processing time (including MFCC feature extraction, OPC and Viterbi decoding) of the system for a 1.0s speech (79 frames) is 0.945s, which is very comparable to 0.877s of the Uni-Lite based system for the same speech. However, in the proposed system, MCU can

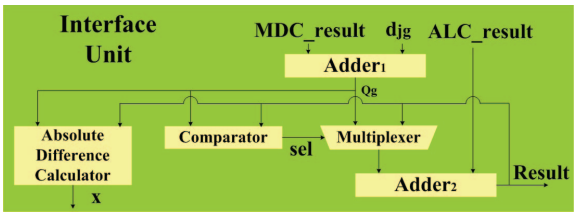

Fig. 6. Block diagram of Interface Unit.

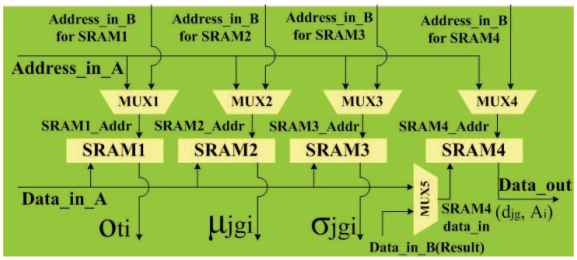

Fig. 7. Block diagram of SRAM Array.

work at 40MHz to fulfill MFCC feature extraction, Viterbi decoding and system control tasks and its power consumption is 10mW. The coprocessor only needs to work at 10MHz and its power consumption is 1.8mW. Table 2 gives the performance comparison between Uni-Lite and coprocessor for OPC only. It can be seen that power consumption of Uni-Lite is 32.5 times that of the coprocessor.

The total dissipated energy of Uni-Lite based system is $0.887 \times 58.5 = 51.89$mW·s. For our system, considering that OPC in coprocessor takes $358 \times 79 \times 9.7 \times 10^{-6} = 0.274$s, MFCC feature extraction in MCU takes 0.242s, and Viterbi decoding in MCU 0.429s, the total dissipated energy (not including data transfer time, which is less than 0.002s) is 7.2mW·s, which significantly reduces energy by 86.2% compared to Uni-Lite based system.

For the dedicated hardware implementation of the entire speech recognition system in [4], the processing time is very small at the expense of 15 times of the hardware resources of our system. Besides, if the number of Gaussian mixtures (G) changes for a new HMM model or a different recognition task, the hardware in [4] should be re-designed. In our system, although the processing time is longer, the architecture of the coprocessor needs no modification when G (or other parameters) changes. Therefore, our system is more flexible and easily adaptable to different models or new recognition tasks. Furthermore, it turns out that for the same HMM model and the same recognition task, for instance the 32-state 1-mixture 38-feature 800-word HMM as used in [4], the proposed system can save energy consumption by 58% compared to the implementation in [4].

5. CONCLUSION

A speech recognition system composed of a MCU and a coprocessor has been designed. The coprocessor is a dedicated design for OPC in CHMM based speech recognition

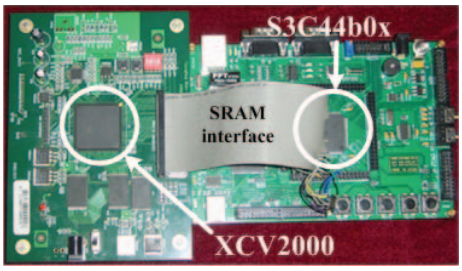

Fig. 8. System implemented by ARM and coprocessor.

Table 2. Performance comparison for OPC for one frame and one state (27-feature 3-mixture word HMM)

	Uni-Lite	Coprocessor
Clock frequency	104 MHz	10 MHz
Processing time	22.3 μs	9.7 μs
Voltage	1.8 V	1.8 V
Power Consumption	58.5 mW	1.8 mW

algorithms, while MCU performs the rest of the processing steps. By combining them, a good tradeoff between power consumption and flexibility is achieved compared to other implementations with only GPP or dedicated hardware.

6. REFERENCES

[1] X. Zhu, Y. Chen, J. Liu, R. Liu, "A Novel Efficient Decoding Algorithm for CDHMM-based Speech Recognizer on Chip." *Proc. of IEEE ICASSP*, 2003, pp. 293-296.

[2] M. Yuan, T. Lee, P. C. Ching, Y. Zhu, "Speech Recognition on DSP: Issues on Computational Efficiency and Performance Analysis". *Proc. of IEEE ICCCAS*, 2005, pp. 852-856.

[3] Zh. Yang, J. Liu, E. Chan, L. Guan, Ch. Ching, "DSP-based System-on-Chip Moves Speech Recognition from the Lab to Portable Devices", *www.embedded.com*, Jan 2007.

[4] S. Yoshizawa, N. Wada, N. Hayasaka, Y. Miyanaga, "Scalable architecture for word HMM-based speech recognition and VLSI implementation in complete system". *IEEE Trans. on Circuits and Systems I*, Vol. 53, No. 1, Jan 2006, pp. 70-77.

[5] W. Han, K. Hon, Ch. Chan, T. Lee, Ch. Choy, K. Pun, P. C. Ching, "An HMM-based speech recognition IC". *Proc. of IEEE ISCAS*, 2003, pp. 744-747.

[6] F. L. Vargas, R. D. R. Fagundes, D. Jr. Barros, "A FPGA-based Viterbi algorithm implementation for speech recognition systems". *Proc. of IEEE ICASSP*, 2001, pp. 1217-1220.

[7] P. Lee, M. Dong, W. Liang, R. Liu, "Design of Speech Recognition Co-Processor for the Embedded Implementation". *Proc. of IEEE EDSSC*, 2007, pp. 1163-1166.

[8] J. Pihl, T. Svendsen, M. H. Johnsen, "A VLSI implementation of PDF computations in HMM based speech recognition". *Proc. of IEEE TENCON*, 1996, pp. 241-246.

[9] C. F. Gerald, P. O. Wheatley, "Applied Numerical Analysis", *Addison-Wesley*, 2004.

[10] "S3C44B0X RISC Microprocessor", *Samsung Inc*. 2005.

[11] "Virtex-II 1.5V FPGA", *Xilinx Inc*. 2001.

[12] X. Huang, A. Acero, H. Hon, "Spoken Language Processing: A Guide to Theory, Algorithm, and System Development", *Prentice Hall*, 2001.

[13] "Uni-Lite Device Specification", *Infineon Inc*, 2005.

[14] D. Seal, "ARM Architecture Reference Manual", *Addison-Wesley*, 2001.

[15] G. Kane, J. Heinrich, "MIPS RISC Architecture", *Prentice Hall*, 1991.