A Low-Power LSI Design of Japanese Word Recognition System

Shingo Yoshizawa, Yoshikazu Miyanaga, Naoya Wada and Norinobu Yoshida

Graduate School of Engineering, Hokkaido University

Kita 13 Nishi 8, Kita-ku, Sapporo, 060-8628 Japan
Tel. +81-11-706-6493, Fax. +81-11-706-6494
e-mail : {yosizawa, miyanaga, wada, yoshida}@ice.eng.hokudai.ac.jp

Abstract: This paper reports a parallel architecture in a HMM based speech recognition system for a low-power LSI design. The proposed architecture calculates output probability of continuous HMM (CHMM) by using concurrent and pipeline processing. They enable to reduce memory access and have high computing efficiency. The novel point is the efficient use of register arrays that reduce memory access considerably compared with any conventional method. The implemented system can achieve a real time response with lower clock in a middle size vocabulary recognition task (100-1000 words) by using this technique.

1. Introduction

Recently technologies of speech recognition methods and circuit design have considerably progressed and consequently a speech recognition chip has been developed with a low-power and small-size circuit.

In previous works, dedicated hardware architectures for HMM based speech recognition were introduced in [1] and [2]. In [1], a dynamic circular fix-point format was proposed to reduce memory bandwidth in output probability calculation of continuous mixture HMM. A FPGA based Viterbi algorithm in discrete HMM was reported in [2]. However, low-power VLSI design in these designs has not been presented yet. As other methods, DSP and MPU can meet the performance requirements in current CMOS process. Due to the increase of recognition words and requirement of high accuracy performance, however a high-speed clock is required.

This paper reports a parallel architecture in HMM based speech recognition for realizing a low-power system. The architecture is intended to decrease total access value between memory and processor. This method is effective in utilizing low-power RAM and operating at lower clock frequency on processor. The following describes the dedicated hardware system, reducing memory access, chip implementation and evaluation of power consumption.

SPEECH RECOGNITION SYSTEM

Recognition procedure can be split into speech analysis and speech recognition. Figure 1 shows a flowchart of the speech recognition system.

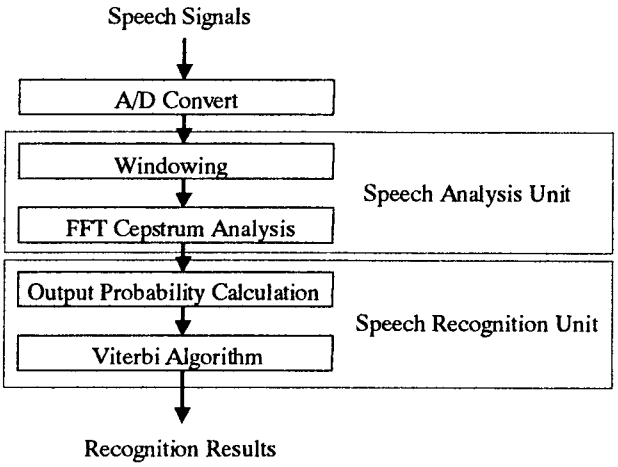

Figure 1. Flowchart of speech recognition system

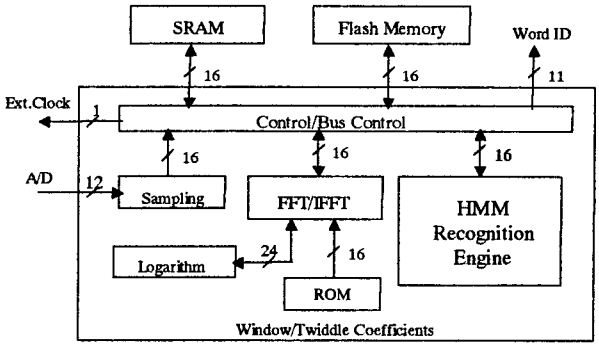

Figure 2. System structure

In a speech analysis unit,windowing and FFT cepstrum analysis [3] are used to extract speech features, which are called input vectors. The features are computed on frame by frame basis. FFT/IFFT circuit, logarithm circuit and window/twiddle coefficients ROM execute the analysis processing, as are depicted in Figure 2. In a speech recognition unit, HMM Recognition Engine is treated as a processor and executes output probability and likelihood score calculations by using Viterbi algorithm [4]. A main task of this recognizer is isolated word recognition in

middle size vocabulary (100-1000 words) based on HMM word models. The system has I/O ports for external SRAM and Flash Memory. SRAM is used to compute FFT cepstrum coefficients and write input vectors. Flash memory stores HMM training data calculated by software systems in advance.

3. REDUCING MEMORY ACCESS

We explain data transfer rates between a memory and a processor in the recognition process. The output probability calculation and the likelihood score calculation are described as the following equations.

(a) Output probability

$$\log b_j(o_t) = w_j + \sum_{p=1}^{P} s_{jp}(o_{tp} + u_{jp})^2 \quad (1)$$

(b) Likelihood score (Viterbi algorithm)

$$\delta_1(j) = p_j + \log b_j(o_1) \quad (2)$$

$$\delta_t(j) = \min_{i=j-1,j}[\delta_{t-1}(i) + a_{ij}] + \log b_j(o_t) \quad (3)$$

$$P^* = \min_{1 \le j \le N}[\delta_T(i)] \quad (4)$$

In (1), o_{tp} is a factor of P-dimensional input vectors. The values u_{jp}, s_{jp} and w_j are factors of Gaussian probability density function (pdf) and have been computed in advance. Viterbi algorithm is given by the formulas of (2)-(4). The value a_{ij} is a HMM transition probability and p_j is a initial probability. The number of HMM states is given by N and T is the total number of frames in input vectors. The values u_{jp}, s_{jp}, w_{jp} and a_{ij} are treated as HMM training data.

The output probability calculation is the most computationally expensive part of HMM-based speech recognition [5]. These equations are not complicated. However, it requires a large amount of HMM training data. The number of parameters of HMM training data is given by $(3+2P)N$ and that of input vectors is PT. Assuming that these parameters are N=12, P=16, T=86 (in case a frame length is 11.6ms) and the number of bit width is 16, the data size of one word model is 0.85kbytes in HMM training data, 0.032Kbytes in input vectors and 0.048Kbytes in results of output probability calculation (memory read and write operations are considered).

In the DSP and MPU based recognition systems, a frame synchronous [6] is utilized. This mechanism computes output probability on frame by frame. This method is suitable for sequential processing. However it requires a large amount of HMM training data transfers since the same training data are transmitted to processor every frame. For example, in case of the number of frames T=86, the rates become 0.85(Kbytes) x 86 = 73.1(Kbytes/s). In a 1000 word recognition task, as is given by W=1000, the total data transfer rates between a memory and a processor amount about 77Mbytes/s (see frame synchronous in Table 1).

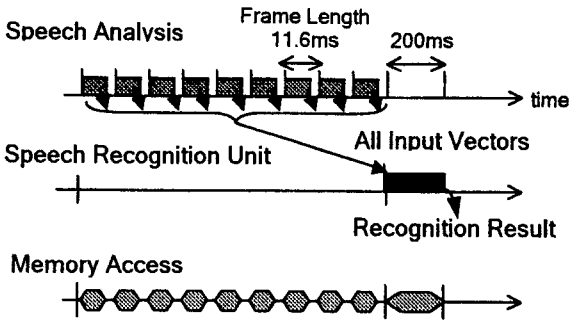

Figure 3. Timing chart in the proposed method

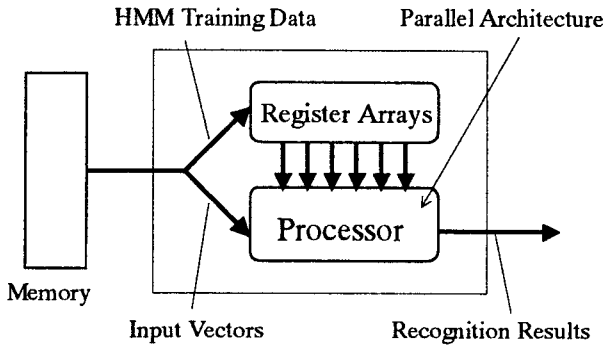

Figure 4. Relationship between memory and processor

To reduce data transfer rates, we have developed the proposed method and its architecture. The key points on the proposed method are described as follows:

- Computing the output probability by using all input vectors (see Figure 3).
- Register arrays which store one word model of HMM training data (see Figure 4).
- Parallel architecture with parallel and pipeline processing for archiving high throughput.

Since HMM training data are independent to the number of frames t, the data stored at internal memory on processor are not needed to transmit to processor repeatedly. The transfer rates can be reduced to $1/T$. However the recognition process begins after finishing the analysis process. The recognition processing time is limited for achieving a real time response. As allowable time is set to 200ms, the data transfer rates can be reduced to 4.2Mbytes/s. Adding 13.76Mbytes/s in input vectors, the total rates amount about 18Mbytes/s (see proposed in Table 1). This method needs higher throughput more than the frame synchronous for the limitation of processing time. To solve this requirement, register arrays and parallel architecture are implemented and they can obtain high computing efficiency. This architecture is described in the next chapter.

Table 1 shows the comparison of data transfer rates between the frame synchronous and the proposed method. The values of input vectors are very small in the frame synchronous since the frame synchronous can equip a small sized internal memory that stores input vectors. The proposed method reduces the rates by about 77% compared with the conventional method.

Required Data Rate (MBytes/s)	Frame Synchronous	Proposed Method
HMM Training Data	$2(3+2P)NWT$ 73.1	$10(3+2P)NW$ 4.2
Input Vectors	$2PT$ 0.003	$10PWT$ 13.76
Output Probability	$4NWT$ 4.13	0 0
Total	**77.23**	**17.96**

Table 1. Comparison of data transfer rates; values are given on condition of N=12,P=16,T=86 and W=1000.

4. PARALLEL ARCHITECTURE

Figure 5 shows the structure of the speech recognition unit. The recognition unit consists of an output calculation unit, a likelihood calculation unit and a recognition decision unit. Multiple port register arrays store HMM training data of one word model and are connected to the output calculation unit and the likelihood calculation unit. In the parallel processing, the implementation of N PEs or P PEs is available by using register arrays. Our system has implemented N PEs since it is convenient for pipeline processing and memory read operations.

The output probability calculation unit is a SIMD architecture and executes addition, multiplication and accumulation in performing fixed-point operations. Its processing throughput amounts from 480 MOPS (at 10MHz) to 2880 MOPS (at 60MHz). Using fixed-point speech recognition simulation and histogram analysis [7] has optimized bit width, saturation and quantization. The fixed-point simulation has measured the number of optimized decimal bits and resulted in 8bit and the histogram analysis has reduced the number of integer bits by the maximum 16bit.

The likelihood calculation unit receives output probability data from the output probability calculation unit every P cycle. The amount of its computation can be decreased to $3N$ per a frame by using left-right HMM. Therefore this processing can finish within P cycle with a few PEs. In the recognition decision unit, a register value is changed if the unit receives a higher likelihood score than the former.

The number of processing cycles in this unit is PTW in the computation and $(3+2P)NW$ in writing data to register arrays, consequently the total number amounts $((3+2P)N+PT)W$. The structure of register arrays is decided by N and P. In the implemented system, these parameter are set to N=12,P=16.

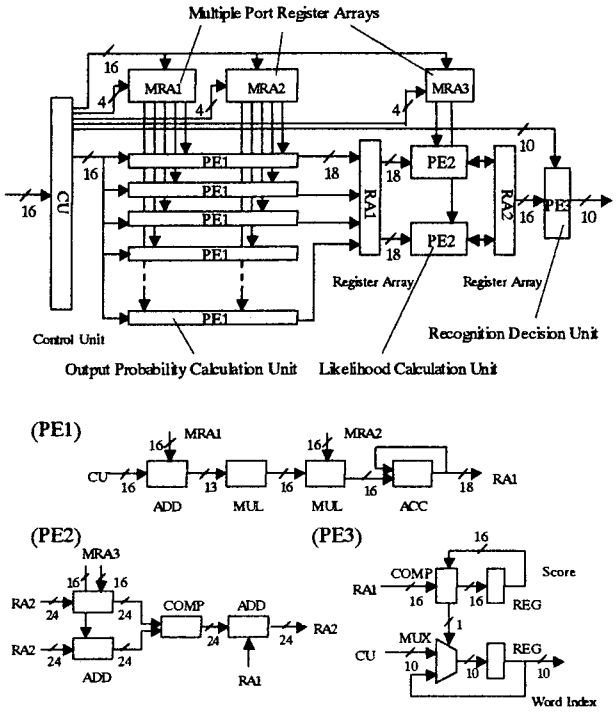

Figure 5. Structure of speech recognition unit (HMM Recognition Engine)

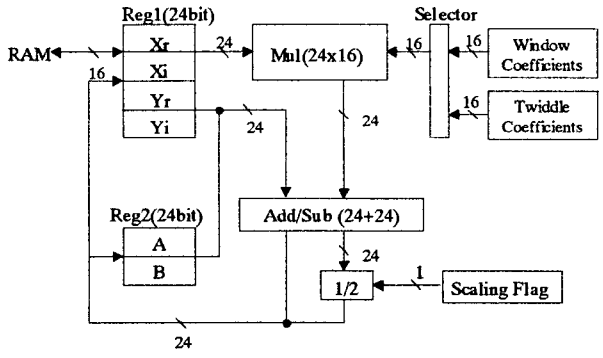

Figure 6. FFT/IFFT circuit

5. SPEECH ANALYSIS UNIT

In the speech analysis unit, the windowing and FFT cepsturm analysis is performed on frame by frame. In this system, sampling rate is 11.025kHz and frame length is 11.6ms. The FFT/IFFT circuit has 24bit fixed point adder and multiplier by using DIT (Decimation in Time) and radix-2 FFT algorithm [8]. This strucutre is shown in Figure 6. The logarithm circuit executes approximate calculation by using STL (Sequential Table Lookup) method [9].

6. CHIP IMPLEMENTATION

We have designed the speech recognition system and implemented a chip using a three metal 0.35um CMOS process and 3.3V power supply. The chip design is supported by VDEC (VLSI Design and Education Center in Tokyo University). The design flow has been based on a Verilog-HDL RTL description. The functional specifications of the chip are listed in Table 2.
The recognition time of one word amounts from 0.18ms/word (at 10MHz) to 0.029ms/word (at 60MHz). Setting the allowable time to 200ms, the 1000 word recognition task can achieve by using 10MHz clock, which is lower frequency. The power consumption has been measured from 93.2mW (at 10MHz) to 567.7mW (at 60 MHz) on condition when the recognition processing is active. Figure 7 shows layout view of the chip.

Technology	0.35um CMOS 3ML
Max. Clock Frequency	63.2 MHz
System Clock Frequency	60, 30, 10 MHz
Chip Size	4.9 mm x 4.9 mm
Gate Counts (2-input NAND)	130k
Voltage Supply(Core)	3.3V
Power Consumption	
60MHz	567.7mW
30MHz	285.2mW
10MHz	93.2mW
Recognition Processing Time	
60MHz	0.029ms/word
30MHz	0.059ms/word
10MHz	0.18ms/word

Table 2. Chip features

7. CONCLUSIONS

We have proposed the parallel architecture that can reduce memory access considerably by using register arrays. It enables to reduce the total access value between memory and processor considerably and have lower clock frequency in the middle size vocabulary recognition task. In the chip implementation, we have reported that power consumption is 93.2mW at 10MHz by using 0.35um CMOS technology and 3.3V power supply.

8. ACKNOWLEDGEMENTS

This research is in parts supported by Semiconductor Technology Academic Research Center (STARC) and VLSI Design and Education Center in the University of Tokyo University (VDEC) in Japan.

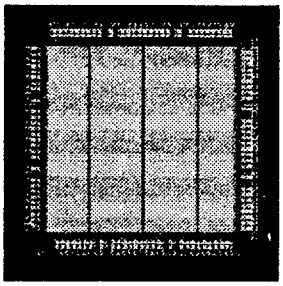

Figure 7. Layout view

9. REFERENCES

[1] Johnny Pihl, Torbjorn Svendsen and Magne H. Johnsen, "A VLSI implementation of pdf computations in HMM based speech recognition," Proc. IEEE Region Ten Conference on Digital Signal Processing Applications, Nov 1996.
[2] Fabian Luis Vargas, Rubem Dutra Ribeiro Fagundes and Daniel Barros Junior, "A FPGA-based Viterbi algorithm implement for speech recognition systems," ICASSP2001, May 2001.
[3] Gray A. H. and Markel J. D., "Distance measures for speech processing," IEEE Trans. Acoustics, Speech, and Signal Processing, ASSP-34, 1, pp.52-59, 1986.
[4] Lawrence R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," Proc. IEEE, 77(2), pp.257-285, Feb 1989.
[5] Shigeki Sagayama and Satoshi Takahashi, "On the use of scalar quantization for fast HMM computation," ICASSP95, vol.1, pp. 213-216, 1995.
[6] Chin-hui Lee and Lawrence R. Rabiner, "A frame-synchronous network search algorithm for connected word recognition," IEEE Trans. on ASSP, vol.37, No. 11, Nov 1989.
[7] S. Yoshizawa, Y. Miyanaga and N. Yoshida, "Design of a parallel and concurrent LSI system for hidden markov models," 16-th Digital Signal Processing Symposium, Nov 2001.
[8] Ediz Cetin, Richard C. S. Morling and Izzet Kale, "An integrated 256-point complex FFT processor for real-time spectrum analysis and measurement," IEEE Proc. Instrumentation and Measurement Technology Conference, pp. 99-101, 1997.
[9] Chen, T. C. "Automatic computation of exponentials, logarithms, ratios and square roots," IBM J. Research and Development, Vol.16, No.4, pp. 380-388, July 1972.