A 40-NM 54-MW 3×-REAL-TIME VLSI PROCESSOR FOR 60-KWORD CONTINUOUS SPEECH RECOGNITION

Guangji He, Yuki Miyamoto, Kumpei Matsuda, Shintaro Izumi, Hiroshi Kawaguchi, and Masahiko Yoshimoto

Kobe University, Kobe, 657-8501 Japan
achilles@cs28.cs.kobe-u.ac.jp

ABSTRACT

This paper describes a low-power VLSI chip for speaker-independent 60-kWord continuous speech recognition based on a context-dependent Hidden Markov Model (HMM). We implement parallel and pipelined architecture for GMM computation and Viterbi processing. It includes a 8-path Viterbi transition architecture to maximize the processing speed and adopts tri-gram language model to improve the recognition accuracy. A two-level cache architecture is implemented for the demo system. The test chip, fabricated in 40 nm CMOS technology, occupies 1.77 mm × 2.18 mm containing 2.98 M transistors for logic and 4.29 Mbit on-chip memory. The measured results show that our implementation achieves 25% required frequency reduction (62.5 MHz) and 26% power consumption reduction (54.8 mW) for 60 k-Word real-time continuous speech recognition compared to the previous work. This chip can maximally process 3.02× and 2.25× times faster than real-time at 200 MHz using the bigram and trigram language models, respectively.

Index Terms— 40 nm VLSI, large vocabulary continuous speech recognition (LVSCR), 3×

1. INTRODUCTION

Speech recognition has been widely used in various applications especially the mobile system, the ubiquitous system and robotics as a human interface. High-end personal computers can accommodate speech recognition tasks well even with large acoustic and language models [1]. However, such software-based methods are not applicable for mobile systems while considering the physical size and power consumption [2]. Additionally, they are unsuitable for next-generation applications such as audio mining, which request the recognizer to deliver results at rates that are 10×, 100×, faster than real-time [3, 4]. Hardware implementation by VLSI or an FPGA is a good approach to satisfy these demands because of its good processing speed and power consumption. Lin et al. reported a Multi-FPGA implementation for 5 k-word continuous speech recognition [5] that achieves 10× faster than real time, but the system is not extendable for larger vocabularies because it is not cost-effective. It needs two FPGAs and two DDR2 DRAMs each with a 64-bit wide data-path. Yoshizawa et al. proposed a scalable architecture for speech recognition [6]. Their chip can have an adjustment between vocabulary size and processing speed, but the system only offers real-time performance with a limited vocabulary of 800 words. Choi et al. developed FPGA and VLSI implementations for 20 k-word speech recognition [7, 8]. They implemented special memory interfaces for several parts of the recognition engine to apply optimized DRAM access, which improves the data transfer efficiency, but the numerous external DRAM accesses cause high IO frequency, which requires a high supply voltage and causes high power consumption in both the FPGA side and DRAM side.

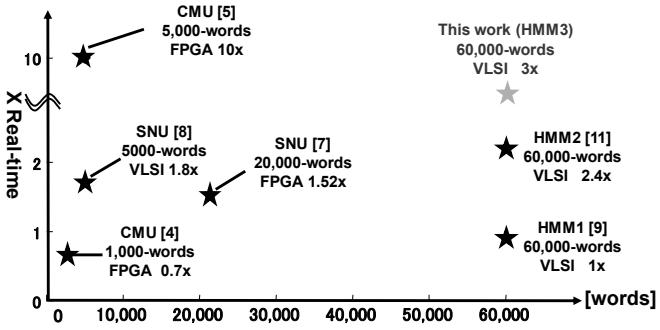

Fig. 1 Vocabulary vs speed.

In Image and Video processing system, the DRAM only acts as a buffer between camera and chip, the pixels are read from DRAM orderly and saved to the on-chip memory. However, in large vocabulary continuous speech recognition (LVCSR) system, the DRAM functions as a data-base which saves the dictionary parameters and language models because they are too large to be stored in the internal SRAM. These data will be accessed randomly during the processing. Due to the characteristics of DRAM , there are several cycles of latency caused by pre-charge and

978-1-4673-6238-2/13 $31.00 © 2013 IEEE

address-setup every time before we read from the DRAM, therefore if the required data are not saved sequentially, the access-efficiency is bad. As a result, speech recognition needs much higher IO frequency than video processing to get the same amount of data from DRAM. Especially, with the number of vocabulary increase, the external memory bandwidth become enormous which causes two problems, firstly, real-time processing is impossible to be achieved because of the I/O frequency limitation. Secondly, large amount of power is consumed by I/O because of the high supply voltage (3.3V). Consequently, reducing external memory bandwidth is one of the most important things to implement a low-power speech recognition system.

In the prior work [9, 10], we presented a VLSI processor (HMM1) for real-time continuous 60-kWord continuous speech recognition. It employs some algorithm optimization and specialized cache architecture. We reduced 95% of the external memory bandwidth and 78% of required frequency. It is the first hardware-based recognizer that can recognize speech in real-time with 60-kWord models. Nevertheless, its processing speed is limited and the internal RAM size reaches 7.8 Mbit, occupying a large area. Afterward, we optimized the on-chip memory and implemented a 4-path Viterbi transition unit in [11] (HMM2) to saved the area and accelerate the processing.

As described herein, to further improve the performance, in this paper, we introduce a 8-path Viterbi transition unit to maximize the processing speed and adopt the trigram language model to improve the recognition accuracy. A two-level cache architecture is implemented for the demo system. We designed and fabricated a VLSI test chip in 40 nm CMOS technology. Results show that the developed chip (HMM3) achieves 25% required frequency reduction (62.5 MHz) and 26% power consumption reduction (54.8-mW) for performing 60 k-Word continuous real-time speech recognition compared to our previous chip HMM2. This chip can maximally process 3.02× and 2.25× faster than real-time at 200 MHz using the bigram and trigram language, respectively. A comparison of the vocabulary size and processing speed among recently announced hardware-based speech recognizers is shown in Fig. 1.

The rest of this paper is organized as follows. The speech recognition algorithm used in this chip is explained in Section 2. Section 3 describes the proposed architecture of the implemented system. Section 4 presents the VLSI implementation and its measurement results. Finally, Section 5 offers concluding marks.

2. ALGORITHEM OVERVIEW

Figure 2 presents the speech recognition flow with the HMM algorithm [12]. **Step 1:** Feature vector extraction: The speech input is sampled using an A/D converter and the mel frequency cepstral coefficients (MFCC) feature vectors are extracted from 30 ms length of speech every 10 ms. **Step 2:** GMM computation: State output probabilities are calculated for all possible sounds that could have been pronounced. **Step 3:** Viterbi Search: $\delta_t(j)$ is calculated for all active state nodes using GMM probabilities, transition probabilities and language models. **Step 4:** Beam pruning: according to the beam width, active state nodes having a higher score (accumulated probability) are selected; the others are dumped. **Step 5:** Output sentence: The word list with the maximum score is output as speech recognition results after final-frame calculation and determination of the transition sequence.

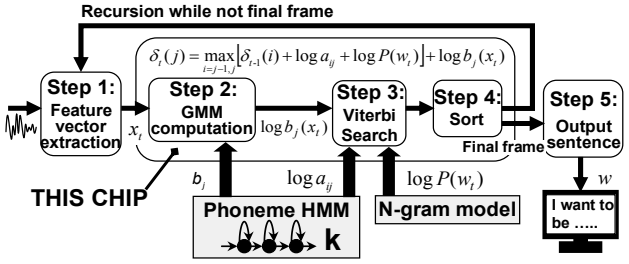

Fig. 2 Speech recognition flow with HMM algorithm.

We calculate the log probability density function (PDF) by its max approximation as shown in Eq. (1).

$$\log b_s(X_t) = \max_m \{C_m - \frac{1}{2}\sum_{d=1}^{D}\frac{(x_d-\mu_{md})^2}{\sigma_{md}^2}\} \quad (1)$$

Therein, $\log b_s(X_t)$ represents the state output probability of a HMM state s for feature vector X_t at time t; x_d stands for the vector component of the feature vector X_t, D is the feature dimension, and C_m, μ_{md}, σ_{md} respectively denote the constant, the mean, and the standard deviation of Gaussian mixture model.

The Viterbi search is divisible into two parts: internal word transition and cross-word transition. Dynamic programming (DP) recursion for the internal word transition is shown in Eq. (2).

$$\delta_t(s_j;w) = \max_{i=j-1,\,j}[\delta_{t-1}(s_i;w)+\log a_{ij}]+\log b_j(x_t) \quad (2)$$

Where a_{ij} is the transition probability from state s_i to s_j, and $\delta_t(s_j;\ w)$ stands for the largest accumulated probability of the state sequence reaching state s_j of word w at time t. Once an internal word transition reach a word-end state, cross-word transition will be treated, the n-gram model is used where the transition probability of a word depends on the n preceding words. We adopt both bigram and trigram for this chip. DP recursion for cross-word transition using bigram is shown in Eq. (3).

$$\delta_t(s_0;w) = \max_v\{\delta_{t-1}(s_f;v)+\log[\,p(w\mid v)]\} \quad (3)$$

Therein, $p(w \mid v)$ stands for the bi-gram probability from word v to word w, s_0 and s_f respectively denote the start state of word w and the last state of word v.

3. ARCHITECTURE

3.1. Speech recognition system

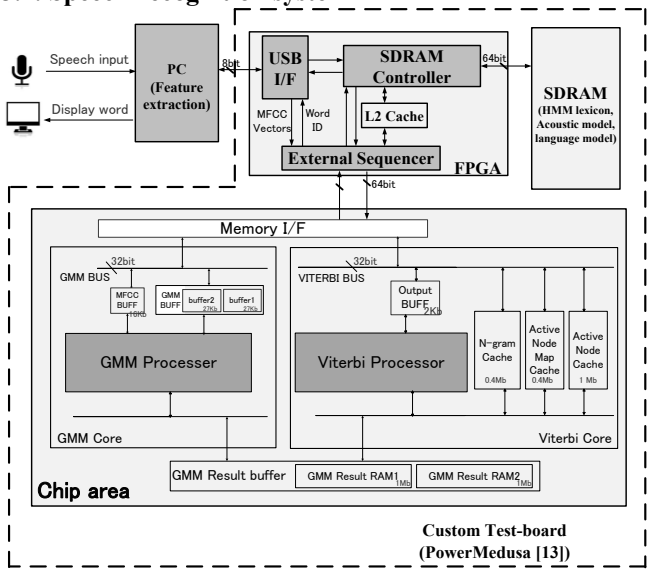

Fig. 3 Overall speech recognition system architecture.

The overall speech recognition system architecture is depicted in Fig. 3, The MFCC feature vectors are extracted using a PC, we separate the feature extraction from the chip because the use of fixed-point computation in the feature extraction part would cause big degradation in recognition accuracy and the computation workload for feature extraction is small thus can be easily handled by PC or an embedded soft-core [4]. The input speech data can either be recorded as an audio stream or with real-time speaking. Before start working, the language and acoustic models are transferred from PC to SDRAM through USB to construct the database. The test chip accesses the DRAM through an on-board FPGA. The data-path of the SDRAM is 64 bit, The data-path for GMM computation (32pin) and Viterbi search (32pin) is separated to support pipeline operation.

A two-level cache architecture is implemented to reduce the latency for accessing SDRAM. The Level-1 cache is the specialized caches we proposed in [9] which are implemented inside the chip and can offer a high hit-rate of 75%. However, when mis-hit occurs, the chip has to access the external SDRAM which causes long latency. As described herein, a Level-2 cache is created in the host FPGA as shown in Fig. 3 . The possible required data are loaded to the L2 Cache during processing. If the required data is found in L1 cache, it will not be transferred to the chip. When the required data is not saved in L1 cache, There's no need to access the SDRAM because the data can be read immediately from the L2 cache.

3.2. 20-frame parallel GMM architecture

We implement a 20-frame parallel architecture for GMM computation to support 3× real-time processing (decided by Viterbi). The GMM core comprise a MFCC buffer for feature vectors, two GMM buffers for loading parameters and 20 GMM computation processors as shown in Fig.4. The same parameters of one mixture are loaded to the registers, then each of the processors computes for one frame. Therefore the parameters are reused by 20 times, which reduce the external memory bandwidth for GMM computation to 1/20. In the previous chip HMM2 [11], we suffered from the pin limitation that only 16 data-pin are available for GMM part. We optimize the pin-placement in this chip by sharing the output pin with Viterbi part because we don't need to output result in GMM computation except the initial test. Therefore the available data-pin for GMM is increased to 32 which is enough to support the required speed . There are two GMM result buffer to support the GMM-Viterbi pipeline operation, each of the result buffers will be accessed by GMM core and Viterbi core respectively during processing.

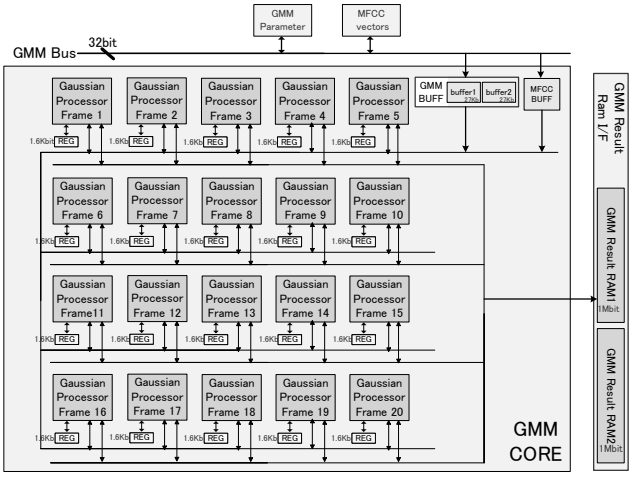

Fig. 4 20-frame parallel GMM architecture.

3.3. 8-path Viterbi transition architecture

The architecture of Viterbi core is shown in Fig. 5. It comprises two active node workspace, an output buffer, a threshold calculator, trellis & token write module and the specialized cache consist of N-gram cache and active node map cache. During transition processing , the active node information and the GMM probabilities can be read from the on-chip memory immediately without miss-hit. However, the N-gram data and active node map data may not be found in the caches. Even the hit-rate of the proposed cache reach 75% [9], the miss-hit still cause big latency because the Viterbi processing will have to stop to wait until the required data is read from the external data-base, which strongly delays the Viterbi processing. Increasing the number of transition-path can hide the miss-hit latency, which improve the processing speed of the

recognition system because the computation time for Viterbi transition is the bottle neck of the GMM-Viterbi pipeline operation [11]. To maximize the utilization ratio of the internal data-path to caches and the external data-path to the SDRAM data-base, we analyze the tradeoff between the number of gates, the number of transition-path and processing speed at 200MHz as presented in Fig. 6. 200MHz is the maximum operating frequency of the test chip. Few speed improvement can be achieved while increasing the number of paths from 8 to 10. This means eight paths is enough to process most of the transitions in pipeline, as caches and external data-base has already been occupied, the extra paths will be in the waiting state most of the time during processing. Consquently, We choose to implement a 8-path Viterbi transition architecture, which offers a processing speed of 3.02×.

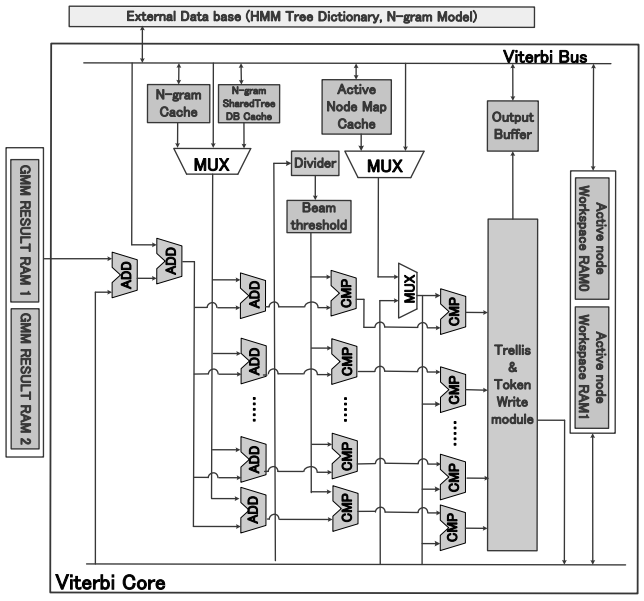

Fig. 5 8-path Viterbi transition architecture.

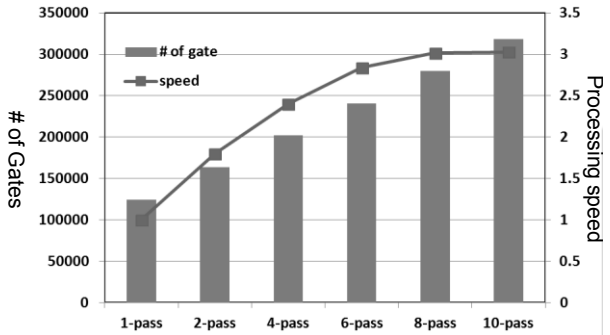

Fig. 6 # of paths versus # of logic elements versus processing speed at 200MHz

Both bigram and trigram are available in this chip. The 60k-word bigram language model consists of 60,001 unigram and 4,000,273 bigram transitions while the trigram language model consists of 60,001 unigram, 2,420 231 bigram and 8,368,507 trigram transitions. To adopt trigram transition, we need to treat much more transitions and remain a large network for the history of the two preceding words as shown in Fig. 7, which is too costly. Therefore we utilize a simplified trigram transition [7] which only consider the best predecessor word. After bigram transition is applied, the best predecessor word can be decided, then the trigram transition from this word is treated. In case of Fig.7, after treating the bi-gram transitions, word B1 is chosen as the best word. Therefore only the tri-gram transitions from B1 is applied, the other tri-gram are not considered. The recognition accuracy for our test patterns is improved by 1.4% when the trigram restrict is added. In that case, the processing speed is decreased to 2.25× (Fig. 8).

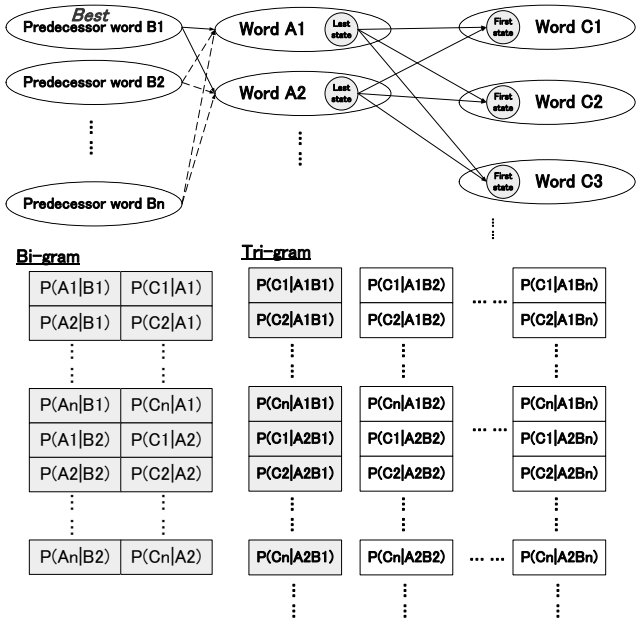

Fig. 7 Cross-word transition using bi-gram and tri-gram.

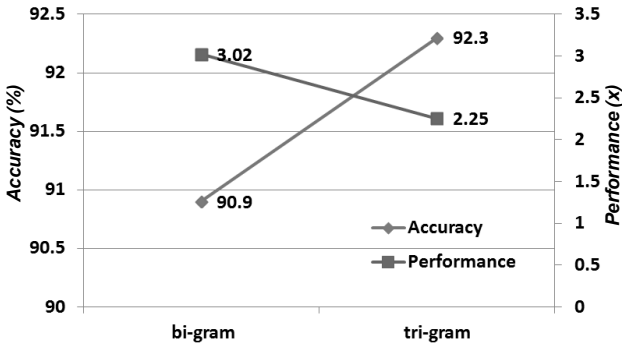

Fig. 8 Performance and recognition accuracy for bigram and trigram (Test speech pattern: 172 Japanese setences).

4. IMPLEMENTATION

We implement a software prototype profiling with Microsoft Visual C++ and a referential hardware using hardware description language (HDL) to check the required memory bandwidth and operating frequency for real-time operation. The required frequency reduction for real-time processing is reduced by 88.9% compared to the base-line system and 25% compared to our previous work HMM2 as shown in Fig. 9.

The layout of the chip, which was fabricated in 40 nm CMOS technology is shown in Fig. 10. It occupies 1.77×2.18 mm^2 containing 2.98 M transistors for Logic and 4.29 Mbit on-chip SRAM. The logic part is placed in the center area and the cache memory is placed around. We evaluated the test chip with a logic tester. The generated Shmoo plot is presented in Fig. 11. The green area of the Shmoo plot shows the available frequency and operation voltage with which the chip can function correctly. 200MHz is the maximum operation frequency of the test chip under the standard operating voltage (1.1V).

Processing speed versus required frequency and the measured power are presented in Fig. 12 and Fig. 13. This chip can process real-time 60-kWord continuous speech recognition with bi-gram model at 62.5 MHz while consumes 54.8 mW and maximally function 3.02× faster than real-time at 200MHz while consumes 177.4 mW. 26% power consumption reduction for real-time processing is achieved compared to HMM2. When tri-gram is used, HMM3 can process real-time operation at 88.9 MHz with power consumption of 76.7 mW and maximally function 2.25× faster than real-time at 200 MHz with power consumption of 165 mW. Table 1 presents a comparison between this chip and some recently announced works in terms of the vocabulary size, GMM model, language model, beam-width, real-time factor, operation frequency, external memory bandwidth , area , logic element and power consumption.

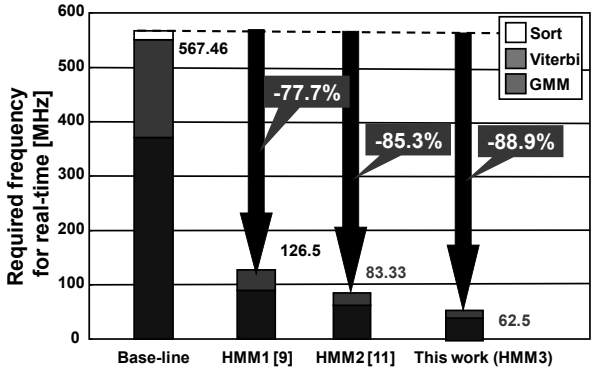

Fig. 9 Required frequency reduction for real-time 60 k-Word real-time continuous speech recognition.

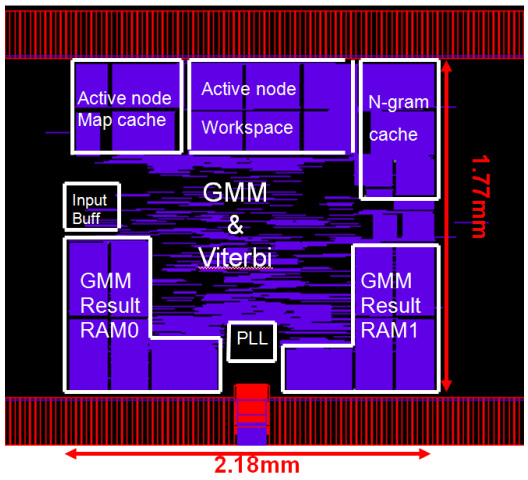

Fig. 10 Chip layout.

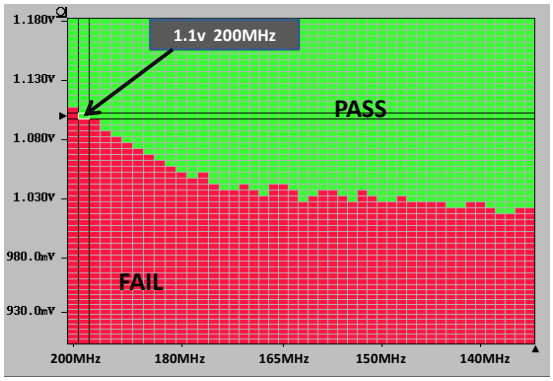

Fig. 11 Shmoo plot generated by a logic tester.

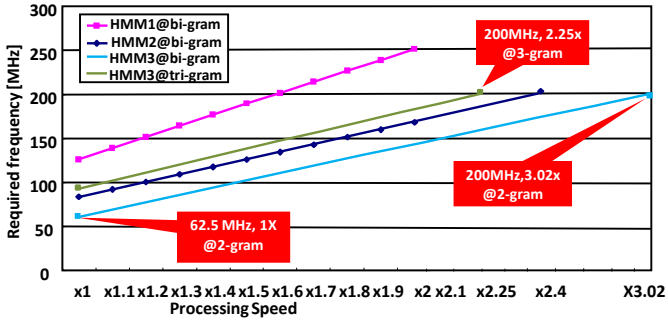

Fig. 12 Processing speed versus required frequency.

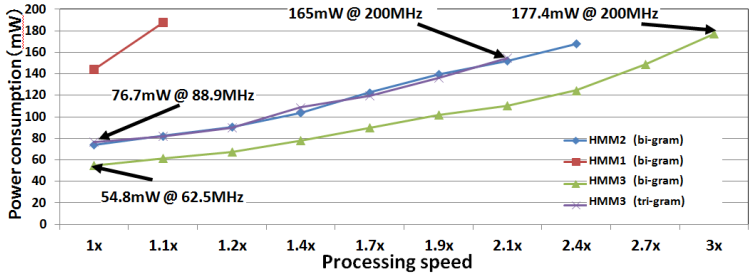

Fig. 13 Processing speed versus Power consumption.

Table 1 Comparison with recently reported works

		[8]	[7]	HMM1 [9]	HMM2 [11]		This work			
Vocabulary (k)		5	20	60	60		60			
Technology		VLSI (0.18 um)	FPGA	VLSI (40 nm)	VLSI (40 nm)		VLSI (40 nm)			
GMM Model	# of states	3,001	3,001	2,000	2,000		2,000			
	# of distributions	16	16	16	16		16			
	# of dimensions	39	39	25	25		25			
LM	# of unigram	5000	19,771	60,001	60,001		60,001		60,001	
	# of bigram	835,000	1,402,259	4,000,273	4,000,273		4,000,273		2,420,231	
	# of trigram	NA	3,617,327	NA	NA		NA		8,368,507	
Viterbi beam width		NA	500	3000	3000		3000	3000	3000	
Real-time factor		0.42	0.66	1	1	0.42	1	0.33	1	0.44
Internal Frequency (MHz)		100	100	126.5	83.3	200	62.5	200	88.9	200
External memory BW (MB/s)		NA	800	70.86	82.6	198	82.6	250	117	264
Power consumption (mW)		NA	NA	144	74.14	168	54.8	177	76.7	165
Core area (mm^2)		15.47	NA	5.5	3.86		3.86			
Logic elements		NA	13,835 slices	1.9 MTr.	2.52 MTr.		2.98 MTr.			
Internal memory (KB)		140.1	416	975	536		536			

5. SUMMARY

We have developed a low-power VLSI chip for 60 k-Word real-time continuous speech recognition. We implement parallel and pipelined architecture for GMM computation and Viterbi processing. It includes 8-path Viterbi transition units and adopts tri-gram search . A two-level cache architecture is implemented for the overall speech recognition system. The measured results show that our implementation achieves 25% required frequency reduction (62.5 MHz) and 26% power consumption reduction (54.8 mW) for 60 k-Word real-time continuous speech recognition compared to the previous work. This chip can maximally process 3.02× and 2.25× times faster than real-time at 200 MHz using the bigram and trigram language models, respectively.

ACKNOWLEDGMENTS

The VLSI chip used in this study was fabricated in the chip fabrication program of VLSI Design and Education Center (VDEC), The University of Tokyo. This development was performed by the author for STARC as part of the Japanese Ministry of Economy, Trade and Industry sponsored "Silicon Implementation Support Program for Next Generation Semiconductor Circuit Architectures".

REFERENCES

[1] A. Lee, T. Kawahara and K. Shikano, "Julius – an open source real-time large vocabulary recognition engine," Proc. European Conf. on Speech Communication and Tech. (EUROSPEECH), pp. 1691-1694, Sep. 2001.

[2] K. Yu, and R. Rutenbar, "Profiling Large-Vocabulary Continuous Speech Recognition on Embedded Device: A Hardware Resource Sensitivity Analysis," Proc. ISCA Annual Conf. of Intl. Speech Communication Association (Interspeech), pp. 995-998, Sep. 2009.

[3] E. C. Lin, K. Yu., R. Rutenbar, and T. Chen, "In silico Vox: Towards Speech Recognition in Silicon" HOTCHIPS 18, August, 2006.

[4] E. C. Lin, K. Yu. R. Rutenbar, and T. Chen, " A 1000-Word Vocabulary, Speaker-Independent, Continuous Live-Mode Speech Recognizer Implemented in a Single FPGA", International Symposium on Field-Programmable Gate Arrays (FPGA), Feb. 2007.

[5] E. C. Lin, and R. A. Rutenbar, "A Multi-FPGA 10x-Real-Time High-Speed Search Engine for a 5000-Word Vocabulary Speech Recognizer," Proc. ACM/SIGDA Intl. Symposium on Field Programmable Gate Arrays (FPGA), pp.83-92, Feb. 2009.

[6] S. Yoshizawa, N. Wada, N. Hayasaka, and Y. Miyanaga, "Scalable architecture for word HMM-based speech recognition and implementation in complete system," Proc. IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 53, no. 1, pp. 70-77, Jan. 2006

[7] Y. Choi, K. You, J. Choi, and W. Sung, "A Real-Time FPGA-based 20,000-Word Speech Recognizer with optimized DRAM Access," IEEE Trans. Circuits Syst. I, Reg. Papers, issue 99, Feb. 2010.

[8] K. You, Y.Choi, J. Choi, and W. Sung, "Memory Access Optimized VLSI for 5000-Word Speech Recognition," JOURNAL OF SIGNAL PROCESSING SYSTEMS, vol.63, no. 1, pp. 95-105, Nov. 2009.

[9] G. He, T. Sugahara, T. Fujinaga, Y. Miyamoto, H. Noguchi, S. Izumi, H. Kawaguchi, and M. Yoshimoto, "A 40 nm 144 mW VLSI processor for Realtime 60 kWord Continuous Speech Recognition," Proc. IEEE Custom Integrated Circuits Conference (CICC), pp.1-4 Sep. 2011.

[10] G. He, T. Sugahara, T. Fujinaga, Y. Miyamoto, H. Noguchi, S. Izumi, H. Kawaguchi, and M. Yoshimoto, "A 40 nm 144 mW VLSI processor for Realtime 60 kWord Continuous Speech Recognition," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 59, no. 8, pp.1656-1666, Aug. 2012.

[11] G. He, T. Sugahara,Y. Miyamoto, S. Izumi, H. Kawaguchi, and M. Yoshimoto, "A 40-nm 168-mW 2.4×-Real-Time VLSI Processor for 60-kWord Continuous Speech Recognition," in Proc. IEEE Custom Integrated Circuits Conference (CICC), Sep. 2012.

[12] X. Huang, A. Acero, and H. W. Hon, Spoken Language Processing-A Guide to Theory, Algorithm, and System Development. Englewood Cliffs, NJ: Prentice Hall, 2001.

[13]http://www.mms.co.jp/powermedusa/concept/index.htl