A 54-mw 3×-real-time 60-kword continuous speech recognition processor VLSI

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

(Citation)
IEICE Electronics Express,11(2):1-9

(Issue Date)
2014-01-25

(Resource Type)
journal article

(Version)
Version of Record

(URL)
https://hdl.handle.net/20.500.14094/90002983

ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.11, No.2, 1–9

A 54-mw 3×-real-time 60-kword continuous speech recognition processor VLSI

Guangji He[a)], **Yuki Miyamoto, Kumpei Matsuda, Shintaro Izumi, Hiroshi Kawaguchi, and Masahiko Yoshimoto**

Department of Information Science, Kobe University, 1–1 Rokkodai, Nada-ku, Kobe, Hyogo, 657–8501, Japan

a) *achilles@cs28.cs.kobe-u.ac.jp*

Abstract: This paper describes a low-power VLSI chip for speaker-independent 60-kWord continuous speech recognition. 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. 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.

Keywords: speech recognition, VLSI, low-power

Classification: Integrated circuits

References

[1] A. Lee, T. Kawahara and K. Shikano: Proc. European Conf. on Speech Communication and Tech. (EUROSPEECH) (2001) 1691.

[2] K. Yu and R. Rutenbar: Proc. ISCA Annual Conf. of Int. Speech Communication Association (Interspeech) (2009) 995.

[3] E. C. Lin, K. Yu. R. Rutenbar and T. Chen: Proc. International Symposium on Field-Programmable Gate Arrays (FPGA) (2007).

[4] E. C. Lin and R. A. Rutenbar: Proc. ACM/SIGDA Int. Symposium on Field Programmable Gate Arrays (FPGA) (2009) 83.

[5] S. Yoshizawa, N. Wada, N. Hayasaka and Y. Miyanaga: IEEE Trans. Circuits Syst. I, Reg. Papers **53** [1] (2006) 70.

[6] Y. Choi, K. You, J. Choi and W. Sung: IEEE Trans. Circuits Syst. I, Reg. Papers **57** [8] (2010) 2119.

[7] K. You, Y. Choi, J. Choi and W. Sung: J. Signal Process. Syst. Signal Image Video Technol. **63** [1] (2009) 95.

[8] G. He, T. Sugahara, T. Fujinaga, Y. Miyamoto, H. Noguchi, S. Izumi, H. Kawaguchi and M. Yoshimoto: Proc. IEEE Custom Integrated Circuits Conference (CICC) (2011) 1.

[9] G. He, T. Sugahara, Y. Miyamoto, S. Izumi, H. Kawaguchi and M.

DOI: 10.1587/elex.10.20130787
Received October 11, 2013
Accepted November 05, 2013
Publicized December 19, 2013
Copyedited January 25, 2014

Yoshimoto: Proc. IEEE Custom Integrated Circuits Conference (CICC) (2012).
[10] X. Huang, A. Acero and H. W. Hon: *Spoken Language Processing-A Guide to Theory, Algorithm, and System Development* (Prentice Hall, Englewood Cliffs, 2001).
[11] "PowerMedusa Custom test board," MMS, Amagasaki, Japan: http://www.mms.co.jp/powermedusa/concept/index.html

1 Introduction

Resently, speech recognition has been widely used in the mobile system, the ubiquitous system and robotics as a human interface. Software-based speech recognition with large vocabulary models [1] can be handled by personal computers but are not suitable for mobile devices while considering the physical size and power consumption [2]. Some server-based distributed speech recognition systems such as "Siri" are utilzed in mobile applications lately, however, delay of such methods strongly depends on the number of users and the signal intensity, therefore they're easily interfered by environment and are not applicable for many high-end applications which need high-speed response and stable performance. Hardware implementation by VLSI or an FPGA is a good approach to satisfy these demands because of its good processing speed and power consumption.

There have been several hardware-based speech recognizers. However, some of them can only support a small vocabulary [3, 4, 5, 7], some of them are limited in speed [3, 5], some others consume much power because of numerous external memory access [4, 6]. In our prior work [8, 9], we employs some algorithm optimization and specialized cache architecture to reduce the external memory bandwidth, which successfully achieves real-time continuous speech recognition with 60-kWord models. 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 ×

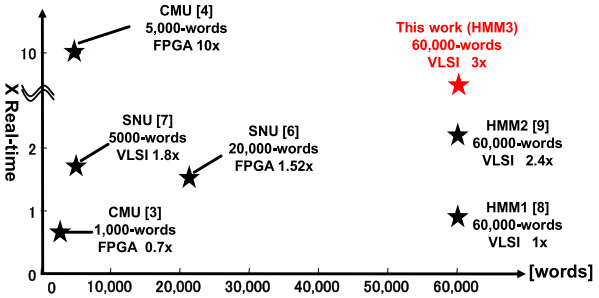


Fig. 1. Vocabulary vs speed.

DOI: 10.1587/elex.10.20130787
Received October 11, 2013
Accepted November 05, 2013
Publicized December 19, 2013
Copyedited January 25, 2014

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.

2 Algorithem overview

Figure 2 presents the speech recognition flow with the HMM algorithm [10]. **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.

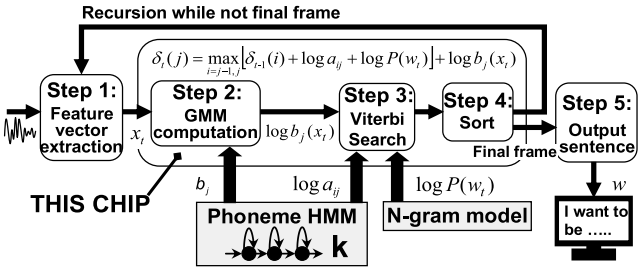


Fig. 2. Speech recognition flow with HMM algorithm.

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

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

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).

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

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

DOI: 10.1587/elex.10.20130787
Received October 11, 2013
Accepted November 05, 2013
Publicized December 19, 2013
Copyedited January 25, 2014

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).

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

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

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.

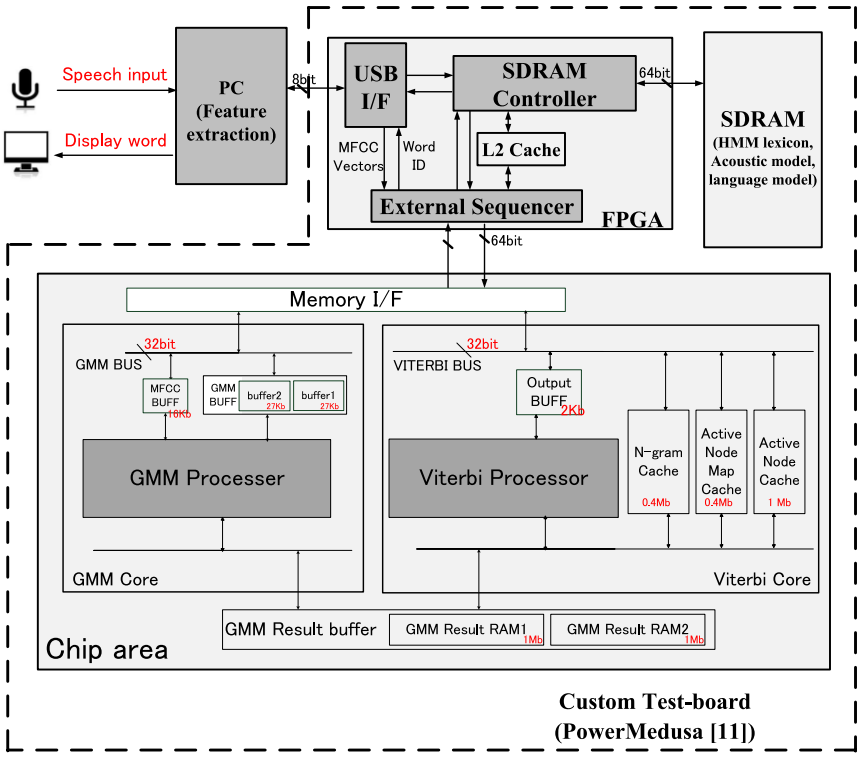
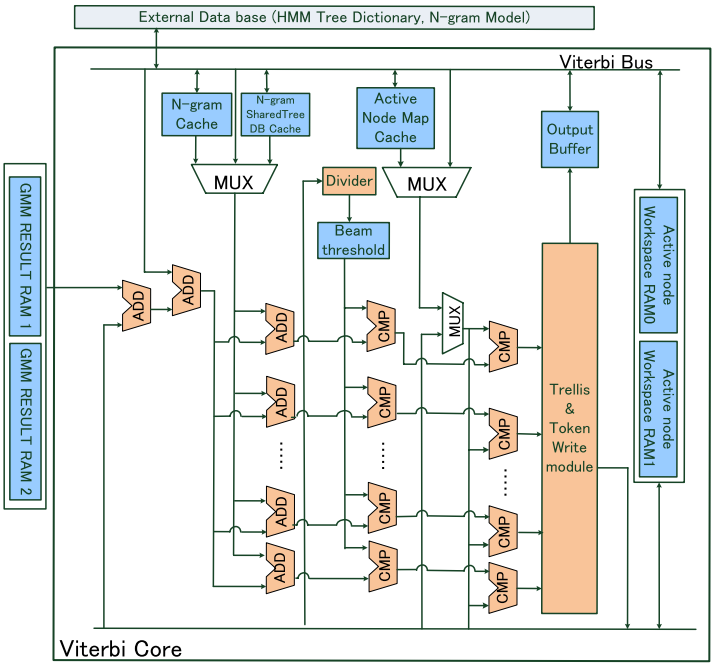


Fig. 3. Overall speech recognition system architecture.

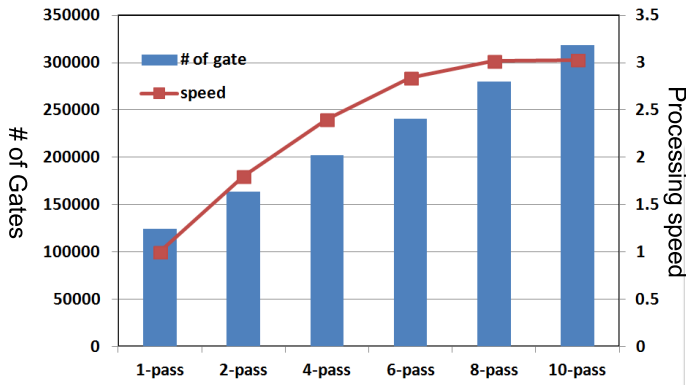
A two-level cache architecture is implemented to reduce the latency for accessing SDRAM. The Level-1 cache is the specialized caches we proposed in [9] which are implemented inside the chip and can offer a high hit-rate of 75%. However, when mis-hit occurs, the chip has to access the external

DOI: 10.1587/elex.10.20130787
Received October 11, 2013
Accepted November 05, 2013
Publicized December 19, 2013
Copyedited January 25, 2014

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. We implement a 20-frame parallel architecture for GMM computation to support 3× real-time processing (decided by Viterbi). 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



(a)



(b)

Fig. 4. (a) 8-path Viterbi transition architecture, (b) # of paths versus # of logic elements versus processing speed at 200 MHz.

DOI: 10.1587/elex.10.20130787
Received October 11, 2013
Accepted November 05, 2013
Publicized December 19, 2013
Copyedited January 25, 2014

Viterbi core respectively during processing.

The architecture of Viterbi core is shown in Fig. 4 (a). 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 200 MHz as presented in Fig. 4 (b). 200 MHz 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×.

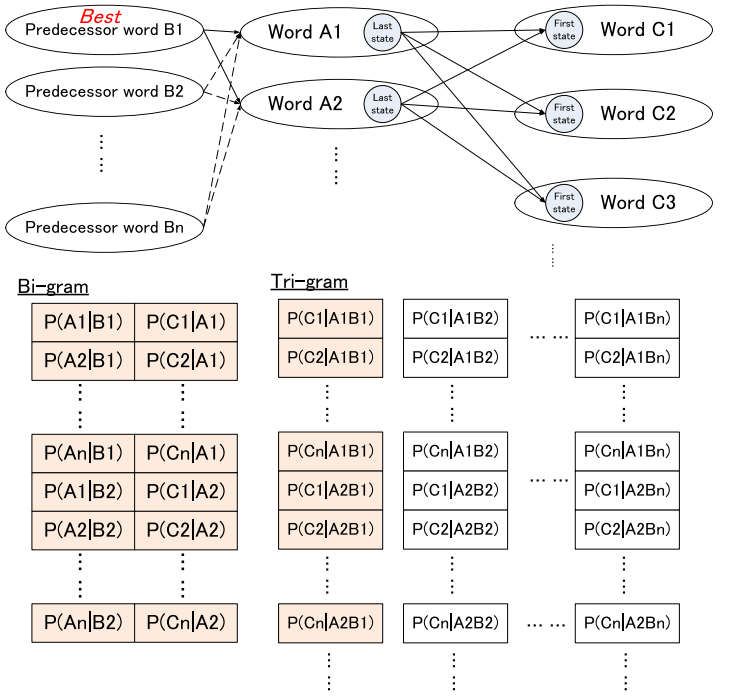
Both bigram and trigram are available in this chip. The 60k-word bigram language model consists of 60,001 unigram and 4,000,273 bigram transitions while the trigram language model consists of 60,001 unigram, 2,420 231 bigram and 8,368,507 trigram transitions. To adopt trigram transition, we need to treat much more transitions and remain a large network for the history of the two preceding words as shown in Fig. 5 (a), 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× as shown in Fig. 5 (b).

4 Implementation

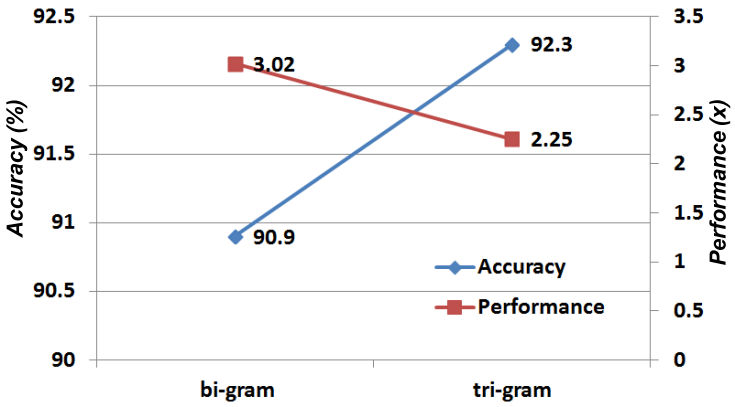
We implement a software prototype profiling with Microsoft Visual C++ and a referential hardware using hardware description language (HDL) to check the required memory bandwidth and operating frequency for real-time operation. The required frequency reduction for real-time processing is reduced by 88.9% compared to the base-line system and 25% compared to our previous work HMM2 as shown in Fig. 6 (a).

The layout of the chip, which was fabricated in 40 nm CMOS

DOI: 10.1587/elex.10.20130787
Received October 11, 2013
Accepted November 05, 2013
Publicized December 19, 2013
Copyedited January 25, 2014



(a)



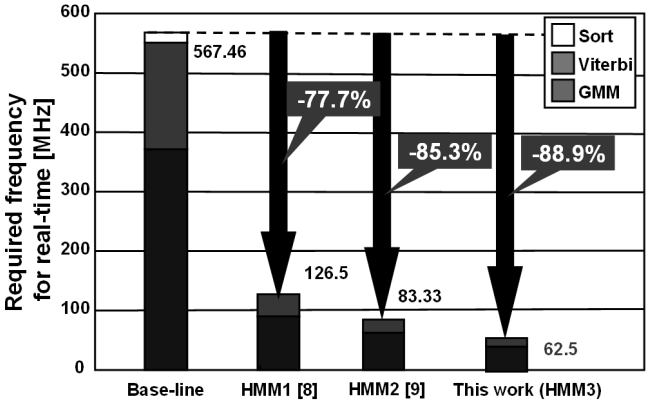
(b)

Fig. 5. (a) Cross-word transition using bi-gram and tri-gram, (b) Performance and recognition accuracy for bigram and trigram (Test speech pattern: 172 Japanese setences).

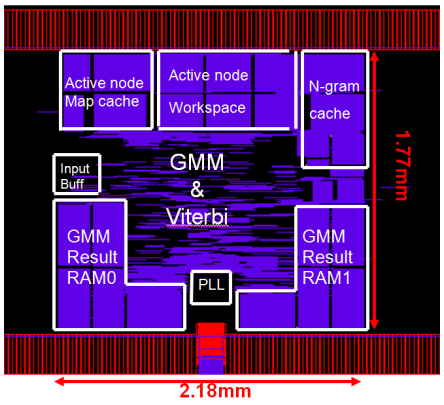
technology is shown in Fig. 6 (b). It occupies $1.77 \times 2.18\,mm^2$ containing 2.98 M transistors for Logic and 4.29 Mbit on-chip SRAM. The logic part is placed in the center area and the cache memory is placed around. We evaluated the test chip with a logic tester. The generated Shmoo plot is presented in Fig. 6 (c). The green area of the Shmoo plot shows the available frequency and operation voltage with which the chip can function correctly. 200 MHz is the maximum operation frequency of the test chip under the standard operating voltage (1.1 V).

Processing speed versus required frequency and the measured power are presented in Fig. 7 (a) and Fig. 7 (b). 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-

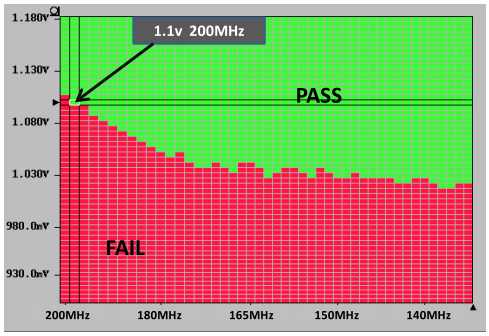
DOI: 10.1587/elex.10.20130787
Received October 11, 2013
Accepted November 05, 2013
Publicized December 19, 2013
Copyedited January 25, 2014



(a)

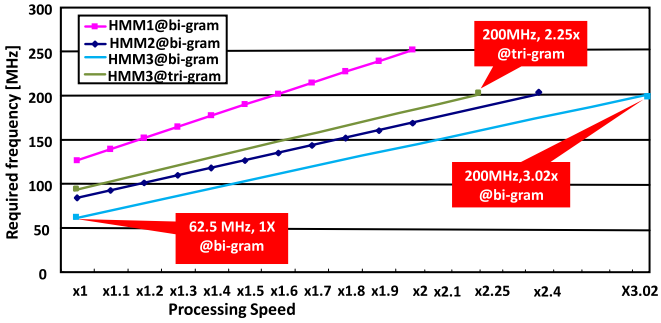


(b)

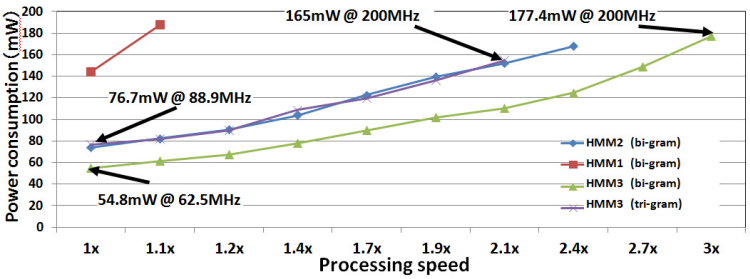


(c)

Fig. 6. (a) Required frequency reduction for real-time 60 k-word continuous speech recognition, (b) Chip layout, (c) Shmoo plot generated by a logic tester.



(a)



(b)

Fig. 7. (a) Processing speed versus required frequency, (b) Processing speed versus Powerconsumption.

DOI: 10.1587/elex.10.20130787
Received October 11, 2013
Accepted November 05, 2013
Publicized December 19, 2013
Copyedited January 25, 2014

time at 200 MHz 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 I 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.

Table I. Comparison with recently reported works.

		[7]	[6]	HMM1 [8]	HMM2 [9]		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 Modu	# 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	2.4	1	3.02	1	2.25
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
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".

DOI: 10.1587/elex.10.20130787
Received October 11, 2013
Accepted November 05, 2013
Publicized December 19, 2013
Copyedited January 25, 2014