A 40 nm 144 mW VLSI Processor for Realtime 60 kWord Continuous Speech Recognition

Guangji He,Takanobu Sugahara,Tsuyoshi Fujinaga,Yuki Miyamoto,
Hiroki Noguchi, Shintaro Izumi, Hiroshi Kawaguchi, and Masahiko Yoshimoto

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

Abstract— **We have developed a low-power VLSI chip for 60-kWord real-time continuous speech recognition based on a Hidden Markov Model (HMM). Our implementation includes a cache architecture using locality of speech recognition, beam pruning using a dynamic threshold, two-stage language model searching, highly parallel Gaussian Mixture Model (GMM) computation based on the mixture level, a variable 50-frame look-ahead scheme and elastic pipeline operation between the Viterbi transition and GMM processing. Results show that our implementation achieves 95% bandwidth reduction (70.86 MB/s) and 78% required frequency reduction (126.5 MHz) for 60-kWord real-time continuous speech recognition. The test chip, fabricated using 40 nm CMOS technology and containing 1.9 M transistors for logic and 7.8 Mbit on-chip memory, occupies 2.2 mm × 2.5 mm area. Measured data show 144 mW power consumption at 126.5 MHz and 1.1 V.**

Keywords—40 nm VLSI, hidden Markov model (HMM), large vocabulary continuous speech recognition (LVCSR)

I. Introduction

A hardware approach for large-vocabulary continuous speech recognition (LVCSR) with implementation by VLSI or an FPGA is demanded especially for use in mobile equipment [1] and intelligent robots because of its advantageous processing speed and power consumption. Lin et al. investigated FPGA implementations for 5-kWord speech recognition [2], but it consumes too much power and is not cost-effective: it requires two FPGAs. Shingo et al. proposed a scalable architecture for speech recognition [3], but their chip uses 136 mW, even with a limited vocabulary of 800 words. Choi et al. investigated 5-kWord and 20-kWord FPGA implementations [4,5]; they implemented a special memory interface for several parts of the recognition engine to apply optimized DRAM access, which reduces the delay for loading data, but they did not reduce the large amount of external DRAM access, which requires power consumption for IO. Ma et al. reported memory-bandwidth reduction of Gaussian Mixture Models (GMM) processing for real-time 20-kWord speech recognition [6], but that method did not accommodate Viterbi processing. Comparison of power consumption among recently announced hardware-based speech recognizers is presented in Fig. 1. To date, the hardware approach has never achieved real-time operation with a 60-kWord language model because of the numerous computation work-load and external memory bandwidth. For low-power and real-time 60-kWord processing, we must reduce both the memory bandwidth and the operating clock frequency.

As a previous work, we described a VLSI architecture for 60-kWord real-time continuous speech recognition in [7]. It employs a cache architecture using the locality of speech recognition, beam pruning using a dynamic threshold, and two-stage language model searching to reduce the memory bandwidth. This paper extends and further optimizes the previous system by application of the following.

- Variable (max. 50) frame look-ahead scheme
- Elastic pipeline operation between Viterbi transition and GMM processing
- Maximize the cache memory hit rate

We analyzed the relationship between accuracy and parameters such as beam-width and the two-stage language model searching frequency. We designed and fabricated a VLSI test chip using 40 nm CMOS technology and measured its performance. Results show that the chip developed for this study can perform 60-kWord continuous real-time speech recognition with power consumption of 144 mW and with little accuracy degradation.

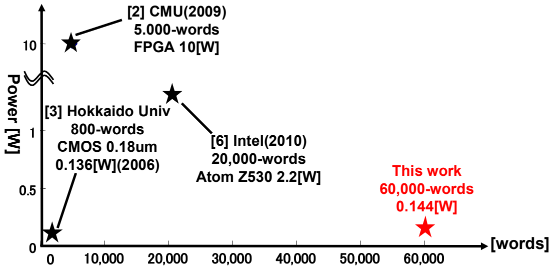

Fig. 1 Comparison of power consumption.

II. Algorithm

2.1 Algorithm overview

Figure 2 presents the speech recognition flow with the HMM algorithm. The following items describe concrete stages. **Step 1:** Feature vector extraction: a feature vector is extracted on a frame-by-frame basis. **Step 2:** GMM calculation: a phonemic-model GMM is read and GMM probability, log $[b_j(x_t)]$, is calculated for all active state nodes. **Step 3:** Viterbi transition: $\delta_t(j)$ is calculated for all active state nodes using GMM probabilities. **Step 4:** Beam

978-1-4577-0223-5/11/$26.00 ©2011 IEEE

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-end state and that having the maximum score is output as a speech recognition result after final-frame calculation and determination of the transition sequence. We profiled referential hardware of Julius 4.0 [8], a well-known Japanese speech recognition system software program using hardware description language (HDL) to a FPGA, with 60-k word speech recognition models with beam-width of 4000. The required memory bandwidth and the required frequency of the prototype necessary to achieve real-time speech recognition respectively reach 3446 MB/s and 567.46 MHz. As described in this paper, we propose several schemes to reduce both of those parameters to realize a low-power, real-time 60-kWord recognition system.

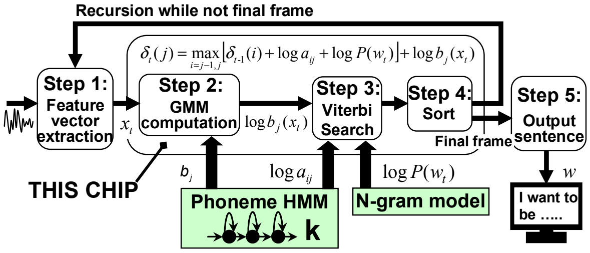

Fig. 2 Speech recognition flow with HMM algorithm computation.

.

2.2 Variable 50 frames look-ahead scheme

The GMM calculations must load numerous parameters, which causes about 576.03 MB/s when processing the 60-k speech recognition. The memory bandwidth can be reduced by sharing the parameter for several frames. Many studies used this look-ahead scheme and they choose to compute the same state for several frames because the state which must be computed in the present frame might need to be computed again in the subsequent frame at high probability. However, it is apparent that the probability decreases when the number of look-ahead frames increases. When a different state is required, the result will become useless. It will cause a delay for the Viterbi operation. For 20-kWord and 60-kWord recognition, it is necessary to maintain sufficient beam width according to the number of words to achieve highly accurate recognition, which is true for almost all states of GMM processing. Therefore, in this study, we compute all 1987 states for the maximum of 50 look-ahead frames. The number of look-ahead frames is variable to make an adjustment between the delay and the memory bandwidth. Using this scheme, the memory bandwidth for GMM processing can be reduced to 13.3 MB/s at most and because that can be accomplished for all states of GMM computation, pipeline operation between GMM and Viterbi is applicable easily.

2.3 Two-stage Language Model Search

We proposed a novel two-stage language model search scheme to reduce the computational work-load and bandwidth for cross-word transitions to isolated trees in [7]. This scheme is derived from the transition frequency difference between phonemic HMM and language HMM. The cross-word transition search is divided into two stages. The first stage is a simplified language model search for the top 10 important transitions of two-gram probability. The second stage is a detailed language model search for all cross-word transitions. As depicted in Fig. 3, in the traditional language model search, only our second search treated every frame. However, in our proposed language model search, the second stage is treated at every *n* frames. By applying this proposed search, the computational amount and memory bandwidth can be reduced to 1/*n*.

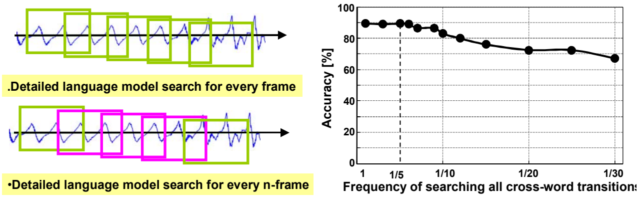

Fig. 3 Two-stage language model search and accuracy degradation.

Figure 3 shows the relation between accuracy and the detailed search cycle in 60-kWord speech recognition in software. Increasing all cross-word search cycles leads to accuracy degradation. The horizontal axis shows the frequency of searching all cross-word transitions. For example, “1/10” frequency means that the detailed language-model search carries at every 10 frames. For this study, we choose to perform a detailed language model search for every five frames to avoid accuracy degradation, thereby achieving 78% reduction in total Viterbi processing.

III. ARCHITECTURE

3.1 Elastic Pipeline Architecture

The overall chip architecture is depicted in Fig. 4. The proposed architecture comprises a global Sequencer, GMM-core, Viterbi-core, and double GMM result buffer to support pipeline operation. Because of the state GMM computation and variable 50 frame look-ahead scheme, it is easy to apply elastic pipeline operation between the Viterbi transition and GMM processing by 1 to 50 frames. Here we will explain why the elastic pipeline architecture can reduce the memory bandwidth for Viterbi transition. Figure 5 shows the memory bandwidth variety of Viterbi transition after applying the two-stage language model search. The frames, when detailed search is used, have the largest amount of data to be read. For conventional operation, these data must be read at 0.01 s, which yields the largest memory bandwidth. By applying this scheme, we can process several frames together, although the frames needing the largest memory bandwidth can use the IDLE time of other frames to load data, which can reduce the peak memory bandwidth by 87.2%. By changing the number of look-ahead frames, we can readily adjust the delay and the memory bandwidth and maximize the elastic pipeline operation efficiency.

3.2 Highly Parallel GMM Architecture

The GMM core has 16 mixture processing blocks, each of which has 1.6 Kb register to preserve the mixture parameter. All blocks are processed simultaneously for the look-ahead frames, which are saved in the MFCC buffer. The parameters will be reused until all look-ahead frames are processed. The mixture results will soon be calculated using the add-log processor based on a look-up table. The mixture computation, add-log calculation, and parameter reading are processed in the pipeline.

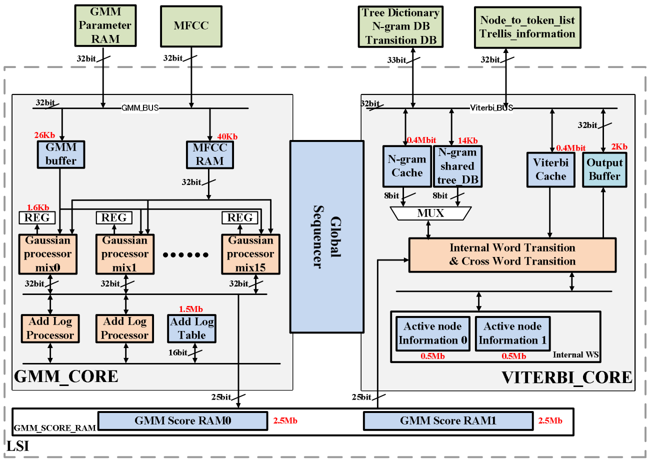

Fig. 4 Proposed speech recognition architecture.

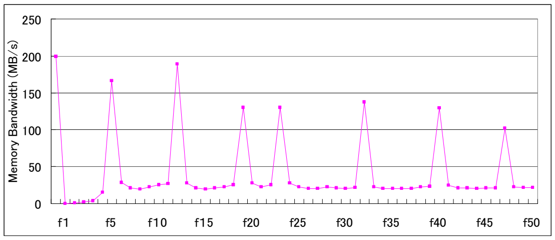

Fig. 5 External memory bandwidth variety of Viterbi transition through frames after the two-stage language model search.

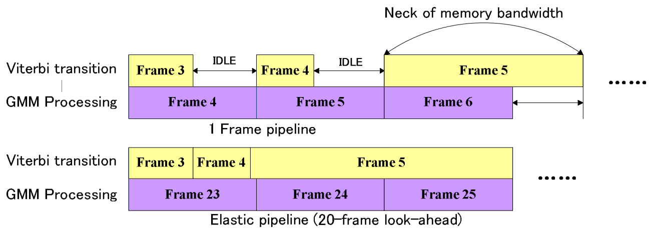

Fig. 6 Elastic pipeline.

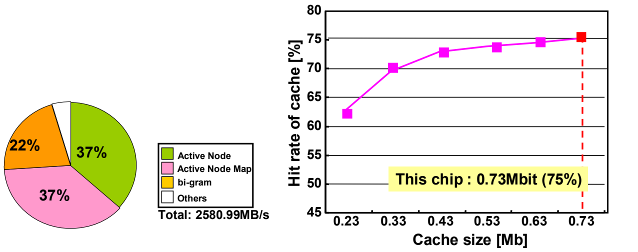

Fig. 7 External memory bandwidth of Viterbi and cache hit rate.

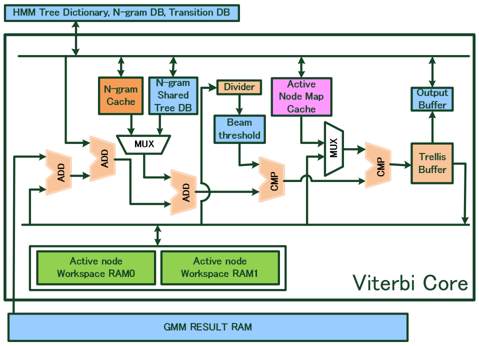

Fig. 8 Viterbi Cache Architecture.

3.3 Viterbi Cache Architecture

Figure 8 shows the Viterbi architecture, for which it is apparent that the Active Node, Active Node Map, and the Bi-gram account for 96% of the memory bandwidth for Viterbi transition, as portrayed in Fig. 7. Therefore, we introduce all active node data and part of the bi-gram and active node map data into the cache memory using the locality of speech recognition that some data which have been used for this frame might be reuse in the following frames. We maximize the cache memory size of bi-gram and active node map to 0.73 Mbit, which can produce a hit rate of 75%. Figure 9 shows the total memory bandwidth reduction by all the proposed schemes. The variable 50 frame look-ahead scheme can reduce memory bandwidth by 16.3% at most, whereas the two-stage language model search and the cache memory can provide reduction of 66.4%, with the search frequency of 5 and cache memory size of 1.73 Mbit.

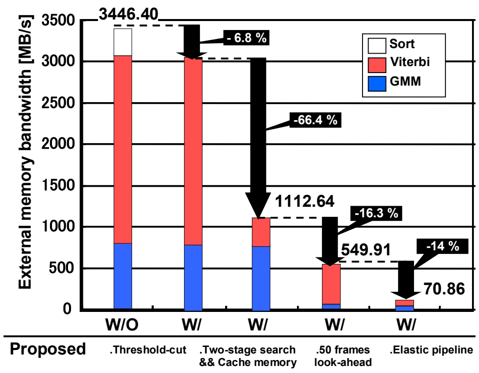

Fig. 9 Bandwidth reduction by the proposed schemes.

IV. IMPLEMENTATION

Figure 10 shows a chip fabricated using 40 nm CMOS technology. It occupies 2.2 × 2.5 mm^2 containing 1.9 M transistors for Logic and 7.75 Mbit on-chip SRAM. The clock gating is implemented in the GMM result RAM and GMM Core.

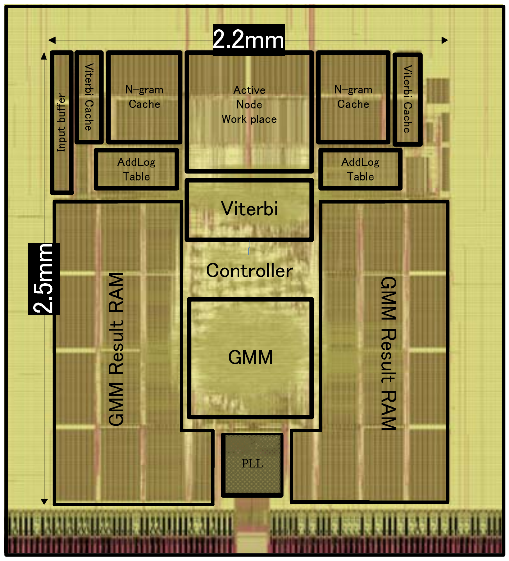

Fig. 10 Chip microphotograph.

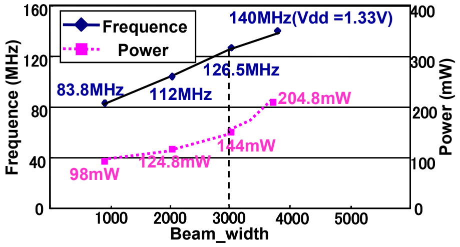

Fig. 11 Measurement results of frequency vs. power consumption vs. beam-width.(Vdd = 1.1V, Vde = 3.3V, Vpll = 1.1V)

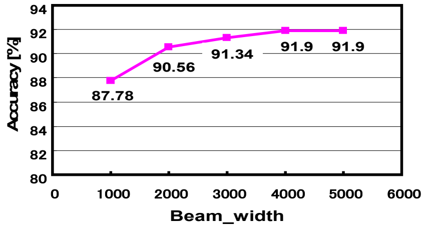

Fig. 12 Beam width vs. accuracy.

Figure 10 shows measured data of power consumption versus operating frequencies versus beam-width. The larger beam-width can yield higher accuracy, as presented in Fig. 12, but it will also increase the computations. This chip can function at 126.5 MHz operation for a beam-width of 3000, although the power consumption is 144 mW with accuracy of 91.34% and 140 MHz for beam-width of 3800. The power consumption is 204.8mW with accuracy of 91.85%.

V. SUMMARY

We developed a low-power VLSI chip for 60-kWord real-time continuous speech recognition. We proposed several schemes to reduce the memory bandwidth and the operating clock frequency. Results show that our implementation achieves 95% bandwidth reduction (70.86 MB/S) and 78% of the required frequency reduction (126.5 MHz) for 60-kWord real-time continuous speech recognition. We fabricated a VLSI test chip in 40 nm CMOS technology and measured the performance. Results show that the chip described in this paper can perform 60-kWord continuous real-time speech recognition at 126.5 MHz with power consumption of 144 mW and with little accuracy degradation.

ACKNOWLEDGMENTS

The VLSI chip 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”.

The authors appreciate valuable discussion shared with Prof. Yasuo Ariki related to the speech recognition algorithm.

REFERENCES

[1] 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.

[2] 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.

[3] S. Yoshizawa, N. Wada, et al. “Scalable architecture for word HMM-based speech recognition and implementation in complete system,” Proc. IEEE Trans. on Circuits and Systems, pp.417-420. Jan.2006

[4] Y. Choi, K. and W. Sung, “FPGA-based implementation of a real-time 5000-word continuous speech recognizer,” Proc. 16th European Conf. (EUSIPCO), Aug. 2008.

[5] 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.

[6] T. Ma and M. Deisher, “Novel CI-Backoff Scheme for Real-Time Embedded Speech Recognition”, *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 1614-1617, Mar. 2010.

[7] H. Noguchi, K. Miura, T. Fujinaga, T. Sugahara, H. Kawaguchi and M. Yoshimoto “VLSI Architecture of GMM Processing and Viterbi Decoder for 60,000-Word Real-Time Continuous Speech Recognition,” Proc. IEICE Trans. on Electronics, Vol.E94-C, No. 4,pp.458-467, April.2011.

[8] 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.