27.2 A 6mW 5K-Word Real-Time Speech Recognizer Using WFST Models

Michael Price, James Glass, Anantha P. Chandrakasan

Massachusetts Institute of Technology, Cambridge, MA

Hardware-accelerated speech recognition is needed to supplement today's cloud-based systems in power- and bandwidth-constrained scenarios such as wearable electronics. With efficient hardware speech decoders, client devices can seamlessly transition between cloud-based and local tasks depending on the availability of power and networking. Most previous efforts in hardware speech decoding [1–2] focused primarily on faster decoding rather than low-power devices operating at real-time speed. More recently, [3] demonstrated real-time decoding using 54mW and 82MB/s memory bandwidth, though their architectural optimizations are not easily generalized to the weighted finite-state transducer (WFST) models used by state-of-the-art software decoders. This paper presents a 6mW speech recognition ASIC that uses WFST search networks and performs end-to-end decoding from audio input to text output.

Algorithms and data structures developed for software speech decoders are also applied in hardware. Fig. 27.2.1 outlines the operation of a speech recognition system, where annotated training data is used to generate statistical models of speech production. A hidden Markov model (HMM)-based Viterbi decoder searches for the most likely path through millions of hidden states. Different statistical models are needed for the transition $p(x_t \mid x_{t-1})$ and emission $p(y_t \mid x_t)$ probabilities. Most modern decoders store transition information in a WFST, which allows the system to combine and optimize several levels of knowledge about the training data from the sub-phonetic to the grammatical level in a single searchable network. The emission probabilities (acoustic model) are represented using a diagonal Gaussian mixture model (GMM), which is well-suited to modeling continuous distributions using relatively few parameters.

Our architecture includes the entire speech-to-text decoding chain and addresses the constraints of low-power embedded systems, including limited on-chip memory capacity and limited off-chip memory bandwidth. The block diagram shown in Fig. 27.2.2 illustrates the flow of information. Audio samples arriving at 16kHz pass through a chain of signal processing elements including an FFT, filter bank, and DCT to generate 39-dimensional mel-frequency cepstral coefficient (MFCC) feature vectors at 100Hz. The filter bank takes advantage of triangular response shapes to generate 26 outputs using only 2 multipliers, as shown in Fig. 27.2.3. We also decimate the real signal into a half-rate complex signal in order to complete the FFT in half as many clock cycles. These operations run at $^1/_{16}$ of the decoder clock frequency and consume an average of 110µW at 0.9V, 50MHz decoder clock.

Each new feature vector triggers a time-synchronous Viterbi search update propagating a set of hypotheses (WFST states with likelihood scores) forward by one time step. A list of hypotheses is read from the active state list (ASL) for the current frame, which is implemented as a hash table in on-chip SRAM. The WFST model is queried for reachable states, resulting in a large set of candidate hypotheses for the next frame. The GMM evaluates the likelihood of observing the feature vector under each of these hypotheses, and likelihood scores are used to select the most promising hypotheses for storage in the ASL for the next frame. Multiplexers swap read and write ports between the two ASLs and the cycle repeats until the end of the utterance. The control module traces backwards through ASL snapshots stored in memory to determine the most likely state sequence, which is converted to a text transcription using a table of WFST output labels.

On-chip memory capacity limits the number of hypotheses that can be stored in the ASL. A constant beam width (search pruning threshold) would have to be set fairly low in order to avoid overflowing the ASL, and desirable hypotheses would be rejected. As shown in [4], the overflow could be stored in off-chip DRAM, but this incurs additional latency and memory bandwidth. By adopting a feedback scheme (shown in Fig. 27.2.4), we are able to obtain a 13.0% word error rate (WER) with an ASL capacity of only 4096 states. We approximate the necessary beam width by regulating the fraction of candidate arcs accepted throughout a Viterbi update. The total number of candidates is predicted by accumulating outgoing arc counts from the previous frame. The histogram of ASL sizes under zero, moderate, and excessive feedback levels is shown on the right in Fig. 27.2.4. With an appropriate feedback gain and clamp range, beam width control will compress (but not eliminate) the natural variation in ambiguity that occurs throughout an utterance. This reduces the workload when there are few good hypotheses, and avoids discarding the best hypotheses when there are too many to be stored in the ASL.

Anticipating the use of slower non-volatile memories to reduce system power, we applied optimizations to reduce memory bandwidth and make memory accesses more sequential. To make related WFST arcs appear close to each other, we order the states in memory according to a breadth-first search. Our fully-associative WFST cache uses the pseudo-LRU eviction algorithm to maximize hit rate and prioritizes arcs at isolated memory locations to reduce page or bank activation penalties. In contrast, acoustic model reads of GMM parameters are highly sequential but can easily exceed 1GB/s; our architecture for reducing this bandwidth to a practical level is shown in Fig. 27.2.5. Repeated access of the same data is avoided by caching the likelihood of each mixture that has been evaluated against a given feature vector; this cache occupies just 98Kb and has an 86% hit rate. We also quantize the GMM means to 5b and variances to 3b, resulting in an 8:1 compression of parameters relative to 32b floating-point format [5]. The impact on decoding accuracy is minimized by selecting a nonlinear quantizer for each parameter according to its empirical distribution in the model. Separate quantization tables are stored for each dimension in order to accommodate nonwhite feature spaces. The combination of caching and parameter compression reduces the GMM memory bandwidth from 2.9GB/s to 54MB/s without requiring storage of GMM results for multiple frames.

This IC was fabricated on a 65nm low-power logic process, and all tests were performed in a real-world demonstration system using an FPGA for external memory access and including all communication latencies. Fig. 27.2.6 shows the measured WER vs. decoding time tradeoff for the Wall Street Journal (Nov. 1992) data set, using a WFST with 2.9M states and 9.1M arcs. The acoustic model contains 10.2M parameters, or 6 times the complexity of that used in [3]. Power consumption is closely correlated with the number of hypotheses evaluated; accuracy and decoding speed can be traded for the desired level of power consumption by adjusting the nominal beam width in conjunction with voltage/frequency scaling. There is no power gating, but we externally gate the clock between utterances for an idle (leakage) power of 42µW. SRAM accounts for 74% of core area and 77% of power consumption, highlighting the importance of memory in speech decoding architectures. Core power consumption averages 6.0mW during real-time decoding at 0.85V and 50MHz. The die photo and specifications are summarized in Fig. 27.2.7.

Acknowledgements:
This work was supported by Quanta Computer, Inc. as part of Project Qmulus, and by an Irwin and Joan Jacobs fellowship. The authors would like to thank the TSMC University Shuttle Program for providing chip fabrication and Xilinx for providing FPGA boards.

References:

[1] J. Choi, K. You, W Sung, "An FPGA Implementation of Speech Recognition with Weighted Finite State Transducers," *IEEE International Conf. on Acoustics Speech and Signal Processing,* pp. 1602–1605, 2010.

[2] J.R. Johnston, R.A. Rutenbar, "A High-rate, Low-power, ASIC Speech Decoder using Finite State Transducers," *IEEE International Conf. on Application-Specific Systems, Architectures and Processors*, pp. 77–85, 2012.

[3] G. He, Y. Miyamoto, K. Matsuda, S. Izumi, H. Kawaguchi, M. Yoshimoto, "A 40-nm 54-mW 3x-real-time VLSI Processor for 60-kword Continuous Speech Recognition," *IEEE Workshop on Signal Processing Systems*, pp. 147–152, 2013.

[4] Y. Choi, K. You, J. Choi, W. Sung, "A Real- Time FPGA-Based 20,000-Word Speech Recognizer With Optimized DRAM Access," *IEEE Trans. Circuits and Systems-I*, vol. 57, no. 8, pp. 2119–2131, 2010.

[5] I.L. Hetherington, "PocketSUMMIT: Small-Footprint Continuous Speech Recognition," *INTERSPEECH*, pp. 1465–1468, 2007.

 978-1-4799-0920-9/14/$31.00

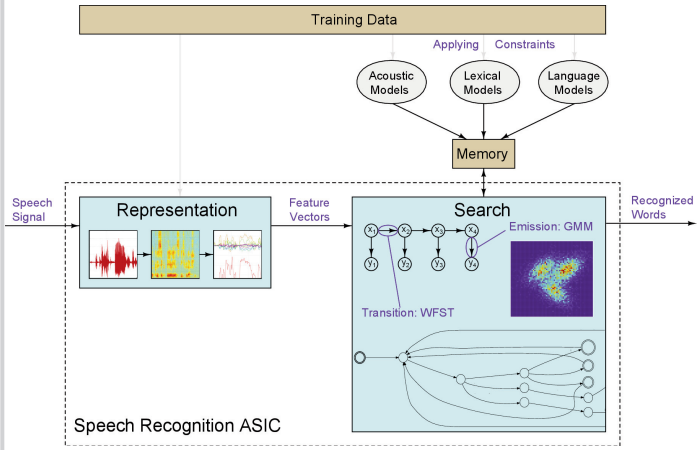


Figure 27.2.1: Overview of speech recognition system; components implemented on chip shown in dashed box.

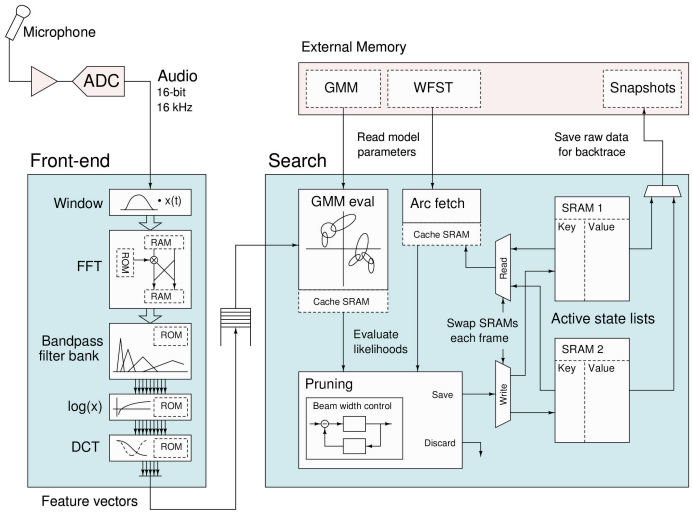


Figure 27.2.2: Block diagram of speech recognition chip.

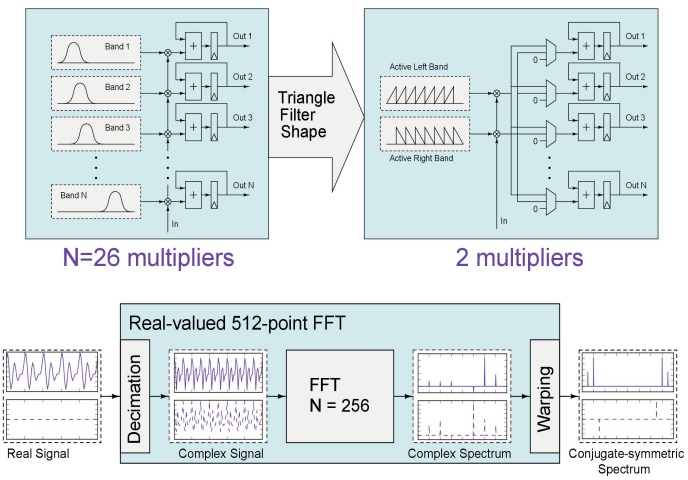


Figure 27.2.3: Bandpass filter bank and FFT optimizations applied to MFCC frontend.

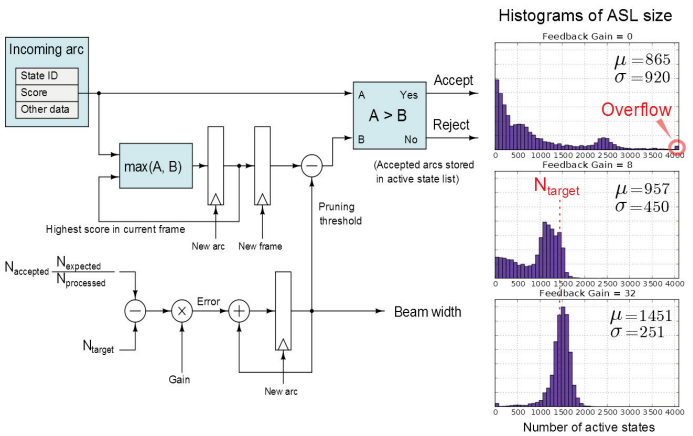


Figure 27.2.4: Architecture and behavior of beam width control via feedback to control variation in number of active states.

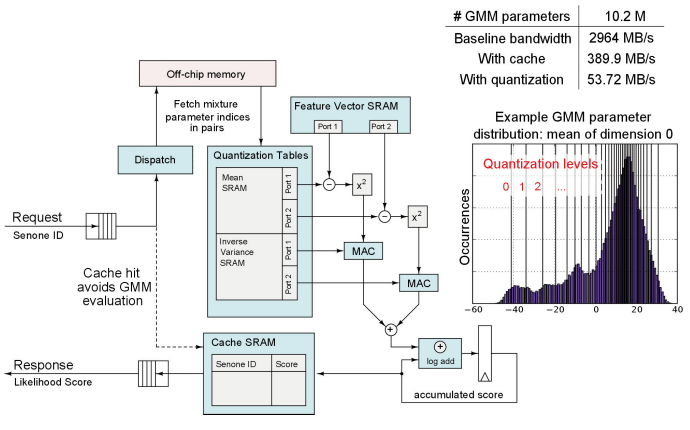


Figure 27.2.5: Quantization and caching of Gaussian mixture models to reduce memory bandwidth.

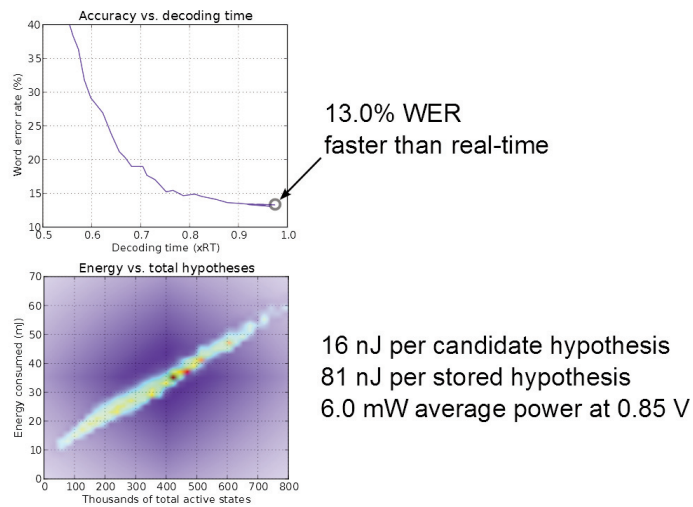
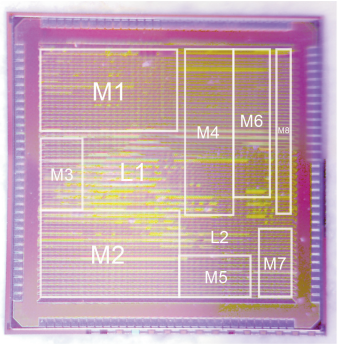


Figure 27.2.6: Word error rate and energy consumption trends at 50 MHz.



Logic regions

L1: Decoder
L2: Frontend

Memories

M1: Active state list 1
M2: Active state list 2
M3: GMM quantization tables and cache
M4: WFST cache data table
M5: Feature vector buffer
M6: WFST cache hash table
M7: Feature and audio log
M8: Frontend and FFT scratch memories

Specification	Value
Process	TSMC 65 nm
Die size	2.5 x 2.5 mm
Package	128-pin LQFP
Logic gates	340k (NAND2 equiv.)
SRAM	2.4 Mb
Supply voltage	0.8 – 1.2 V
Power consumption	5 – 23 mW (core)
Clock frequency	40 – 110 MHz
# WFST states	2.9M
# GMMs	4k x 32 Gaussians

Figure 27.2.7: Die photo and summary of chip specifications.

978-1-4799-0920-9/14/$31.00