A Real-time Architecture for Viterbi Scoring in HMM-Based Isolated word recognition systems

HMM을 이용한 고립 단어 인식 시스템에서의 Viterbi Scoring을 위한 실시간 VLSI구조

Soon Young Yoon*, Hwang Soo Lee*

윤 순 영*, 이 황 수*

ABSTRACT

In this paper, a dedicated architecture is proposed for the Viterbi scoring procedure in hidden Markov model(HMM)-based real-time isolated word recognition systems. The proposed architecture consists of a dual port register file and add-minimize/maximize circuits. By using the tag bits associated with state metrics and model parameters in the external memories, the proposed architecture can incorporate various HMM topologies with ease.

국문요약

본 논문에서는 Hidden Markov Model(HMM)에 기초한 실시간 고립 단어 인식 시스템에서의 Viterbi 알고리즘을 위한 전용 VLSI구조를 제안하였다. 제안된 구조는 듀얼포트 레지스터 파일로 입출력 부하를 줄이고 가산-최소/최대 연산부의 병렬 연산 구조를 이용하여 실시간 동작이 가능하도록 설계되었다. 모델 인자와 상태 변수의 값에 태그들을 덧붙임으로써 이 구조는 대표적인 HMM 구조들을 쉽게 구현할 수 있다.

I. Introduction

During the past decade, researchers have successfully applied hidden markov modeling to robustly represent various units of speech and to recogniz e speech under different constraints[1-3]. Especially the SPHINX system[3] has achieved the recognition rate of 96.2% for a large-vocabulary speaker-independent continuous speech recognition task, making the real-time H/W implementation of the hidden Markov mode(HMM)-based speech recognition more feasible So far not so many real-time systems have been reported for the task[4,5]. In this paper, we are concerned with isolated word recognition based on VQ/HMM and propose a dedicated architecture for the Viterbi scoring procedure The recognition system adopts 49 HMMs that model phoneme-like subword units[6]. A model for a word is constructed by concatenating the constituent HMMs out of the 49 models. The outputs of a bank of 17 critical-band-spaced filters, ranging from 130 to 4300 Hz is used as feature vectors. In our system, a feature vector is obtained every 10 ms. To search the best matching word from a vocabulary of size V, the feature vectors are

*Department of Electrical Engineering KAIST.

at first vector quantized and the VQ codebook indices are used as the observation symbols. The overall block diagram of the isolated word recognition system under consideration is shown in Figure 1

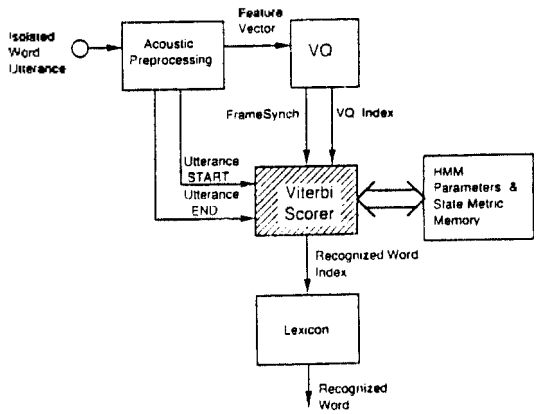


Fig.1. The overall block diagram of the isolated word recognition system.

Given the observation symbol sequence $\{O_1, O_2 \cdots O_t\}$, the following are carried out for each word model λ, $1 \leqq r \leqq V$ in logarithmic Viterbi scoring procedure[1]:

i) initialization : $S_1(1)=b(O_1)$, $S_1(i)=-\infty$, $2 \leqq i \leqq N$,

ii) recursion : for $S_t(j)=\max_{1 \leqq i \leqq N}\{S_{t-1}(i)+a_{ij}\}+b_j(O_t)$, $1 \leqq j \leqq N$,

iii) termination : $P=S_T(N)$,

where N denotes the number of states in the model, and a_{ij} and $b_j(O_t)$ denote the logarithmic versions of the state transition probabilities and the symbol output probabilities, respectively. Let us call $S_t(j)$, a_{ij}, $b_j(O_t)$ as the state metric of state j at t-th recursion, the state transition metric from i-th to j-th state, and the symbol output metric of state j with the input observation symbol O_t, respectively. Multiplications in the original Viterbi scoring procedure are converted to additions and underflow is avoided by taking logarithms of the probabilities.

The viterbi algorithm is a kind of time synchronous forward dynamic pogramming techniques, and equivalent to the shortest path finding problem on the trellis[7]. With the number of the states N and the length of the observation symbol sequence T, the algorithm has the computational complexity of $O(N \cdot T)$ to $O(N^2 \cdot T)$ according to the complexity of the transitions in a trellis stage. Hence with the increase of the input symbol rate or of the number of states, the required amount of computation for real-time processing grows to exceed the capabilities of the general purpose processors. Furthermore since the multiply-accumulate pipelining is not well suited to the repeatedly used add minimize / maximize operations, it seems inefficient to use conventional digital signal processors for the task.

Many of the reported architectures for the Viterbi algorithm focus on the digital communication applications at very high input symbol rates where there is usually only one set of transition probabilities for a trellis stage[8][9]. But, for the Viterbi scoring in the real-time speech recognition based on HMM, during each trellis stage, computations with respect to all the model parameter sets are to be performed at relatively low input observation symbol rates around 100 symbols per second. Since in large-vocabulary recognition systems the number of the model parameter sets usually exceeds 1000, speed limitation due to huge amount of memory acess is a major bottleneck. Whereas architectures for the Viterbi algorithm in digital communication systems are optimised for the types of interconnections such as de Bruijn type and completely bipartite type, in HMMs for the speech recognition a state is usually connected to the three or fewer lower numbered states so that the transition matrix is sparse and upper-diagonal. It is still to be exploited that the task at hand is isolated word recognition rather than

continuous speech recognition. First, the back-tracking procedure doesn't have to be performed. Secondly, the access style of the external memories is sequential. Hence address pins are dispensable and off-chip counters controlled by increment / hold / reset signals alone from the Viterbi scoring chip can generate the addresses.

Ⅱ. THE PROPOSED ARCHITECTURE

Two of the most widely used topologies of HMMs for speech recognition are shown in Figure 2. These models usually represent sub-word units like phonemes, diphones, or triphones. Word models are constructed by concatenating these sub-word units. Here, the symbol output probabilities are assigned to states, but they can be attached to the transition arcs with a minor change to the architecture. In most of the HMMs we are aware of, the states can be numbered such that the difference between the indices of the current state and the preceding states is less than three for most states. Let us call such a transition a *regular* one and otherwise, call the transition an *irregular* one. And a state which all the transition arcs incident on are regular is to be called a *regular* state, otherwise the state is to be called an *irregular* state.

If every state in a HMM is regular as shown in Figure 2.(a), the Viterbi scoring for the HMM will be efficiently carried out in a simple architecture shown in Figure 3.

To cope with the irregular transitions, we propose an architecture shown in Figure 4, using tags and a dual port internal register file. The register file keeps the state metrics that may be used as preceding states for irregular states. Since every irregular transition occurs between states contained in the same phoneme-level HMM the size of the dual port register file can be kept minimal as the maximum number of states among all the phoneme

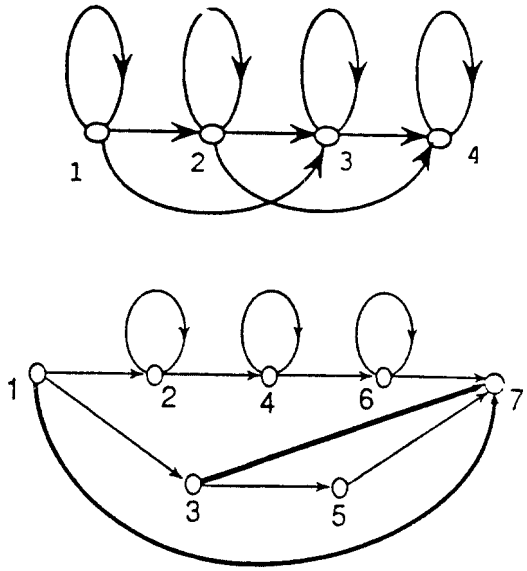


Fig.2. Typical examples of left-to-right HMM topology : (a) a simple regular HMM toplogy, and (b) the HMM topology used in the SPHINX system.

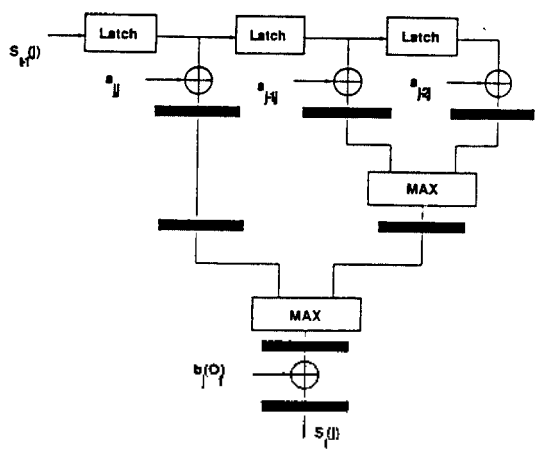


Fig.3. A simple architecture for the HMMs with regular transitions only.

HMMs minus 3. The state transition metrics are stored in four separate memories. Three memories store the regular transition metrics, and the other stores the irregular ones. The symbol output metrics are stored in a separate memory.

Fixed-point unsigned integer format is used. Simulation results show that fixed point operations have comparable performance, and 8-bit unsigned

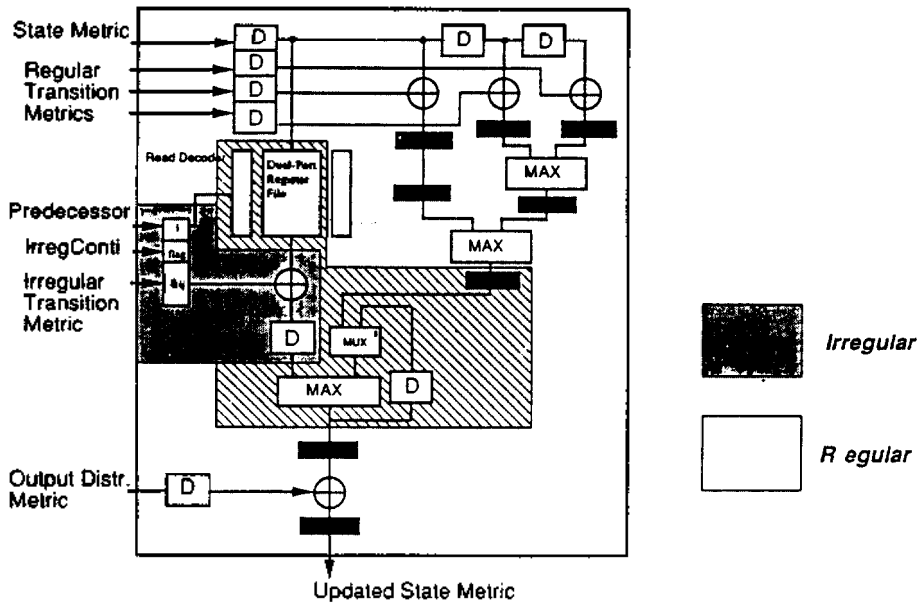


Fig.4. The proposed architecutre for the Viterbi scoring.

integer format for model parameter metrics with 16-bit for the state metrics has proved sufficient. Since due to the reducibilty of the HMM used in most speech recognition systems the magnitude of the terminal state metric increases in linear order as the duration of the utterance gets longer, more than 16 bits for a state metric will be required if the utterance durations are longer than that of our data set.

The architecture is partioned into three differently shaded parts according to their activity spans as shown in Figure 4. The darkly shaded *irregular* part and the lightly shaded *regular* part are held inactive by the following scheme using tagged data formats. Control is performed using tags for the state metrics and the irregular transition metrics together with two utterance end point detection signals and a synchronization signal to indicate an arrival of a VQ index every 10ms. The tags are attached in advance to the metrics as shown in Figure 5. The data format for a state metric includes a 1-bit flag to indicate whether the corresponding state is irregular and a 3-bit concate-

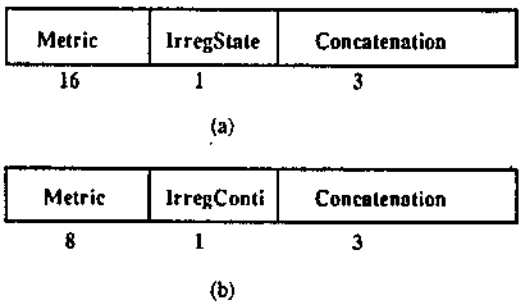
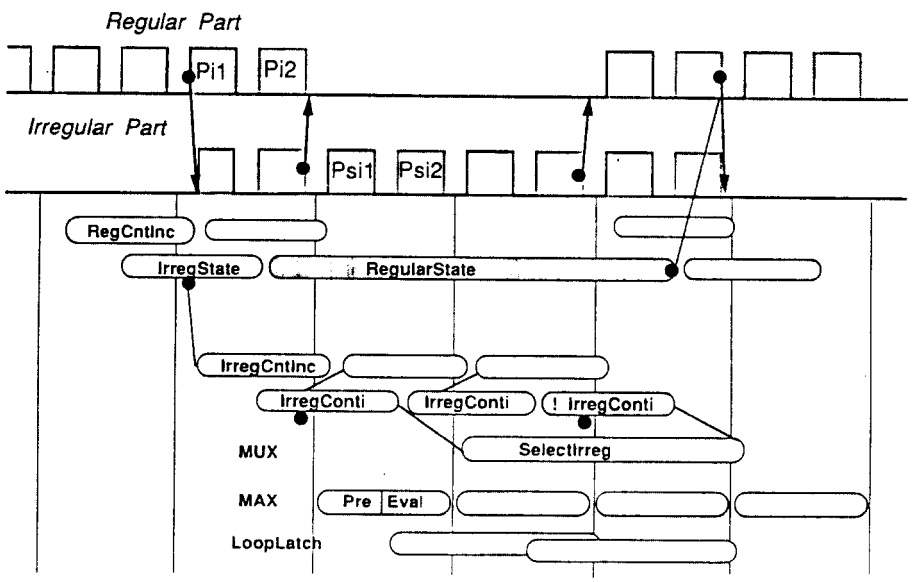


Fig.5. The Tagged data formats for (a) a state metric and (b) a transition metric.

nation information field. According to the 3-bit field, the status of the state is classified into " Not-a-Number", "Phoneme Start", "Word End and Regular", "Word End and Irregular", and " Irregular". A "Phoneme Start" tag is used to indicate the just fetched state metric is associated with the starting state in a phoneme HMM. A 1-bit shift register with the number of stages equal to that of the words in the dual port register file (DPRF) drives the word lines for the DPRF instead of the write address decoder. Provided that the "Phoneme Start" tag is high, the shift register gets initialized to "1 0 0 ··· 0", and on the subs-

equent fetches of state metrics it is right-shifted by a bit, filling the left-most bit with "0". "Word End and Regular" and "Word End and Irregular" tags are used for indicating the current state is the terminal state of a word HMM to help select the word that scores highest at the last VQ index frame. An "Irregular" tag indicate the currently fetched metric is for an irregular state and an irregular transition metric is to be fetched and added to the metric of an incident state and compared and selected In the irregular transition metric memory the transition metrics are stored in ascending order of the index j of the irregular states and with the same index j, in ascending order of index of incident states on the state j where the indices are as defined for the phoneme HMMs. Together with an irregualar transition metric, the index i for the incident state is also stored to be used as the read address for the DPRF.

The flag associated with the transition metric indicates whether there still remains another incident state on the current state. All parts of the processing element except for the parts relevant to process the irregular transitions are stalled if an irregular state have more than one irregular transitions to itself. The details of this control scheme are shown in Figure 6, where an irregular state with three irregular transitions incident on it is taken as an example. Along with the timing diagram, control signals are also shown. We assume typical two-phase clocks Phi1 and Phi2 for the precharge-evaluate dynamic CMOS technology, and Pi1 / 2 and Psi1 / 2 are stalled Phi1 / 2s, which activate the *regular* part and the *irregular* part respectively according to the control scheme. The abridged names for the control signals represent the following. "Irreg CounterInc" increments



2-Phase Clock		Phi1, Phi2
IrregCounterInc	⇑	Phi1 & [IrregState V IrregConti]
StallRegular	⇓	Psi2 & IrregConti
ReleaseRegular	⇓	Psi2 & ! IrregConti
StallIrreg	⇑	Pi1 & ! IrregStateDelay_1
ReleaseIrreg	⇑	Pi1 & IrregState
SelectIrreg	⇓	Psi2 & IrregContiDelay_1

Fig.6. An example of control timing and signals.

the counter which generates the addresses for the irregular transition metric memory : "IrregState" indicates whether the state metric data word in the input state metric latch is irregular : "IrregConti" indicates whther there still remains another irregualr transition metric incident on the current state to be fetched : the suffix "Delay-1" means one-clock delayed versions : the "StallRegular" signal deactivates Pi1 / 2 clock and the "ReleaseRegular" signal reactivates the inactive Pi1 / 2 clock, and "StallIrreg" and "ReleaseIrreg" act similarly for Psi1 / 2 clock : "RegCntInc" increments the counters for the state metric memroy and the three regular transition memories : "&", "V", and "!" represent logical "and", "or", and "negation" respectively : upper and lower arrows represent rising edge and falling edge triggerings respectively : "SelectIrreg" is the select signal for the MUX for the computation of the state metric with more than two irregular transitions on it. In the timing diagram, each round box stands for a valid unchanged duration of the value of the corresponding signal. For instance, "ReleaseIrreg" is asserted active in the rising edge of Pi1 only if the state associated with the just fetched state metric is irregular. In Figure 6, the irregular state with three irregualr impinging transitions causes the regular part stalled for two clock cycles. If an irregualr state has only one irregular transition upon it, "StallRegular" will not be asserted and so no extra cycles will be needed.

Ⅲ. Discussions and Further Studies

The proposed architecture can incorporate various HMM topologies by using tags and flags. The machine cycle of the architecture is determined to be equal to the maximal memory read cycle dut to limitation from memory access bottlenck. The architecture updates a regular state metric every macnine cycle by pipelining and takes extra cycles to update a irregular state with more than one irregualr transitions incident on it. The updated state metric is written into the state metric memory. A dual port RAM is used as the state metric memory. For the topology of HMMs used in the SPHINX system, only the terminal state takes an extra cycle. If a vocabulary consists of words whose HMMs have 50 states on the average in a word model and almost all states have less than 2 irregular transitions incident upon them, the architecture can recognize in real-time up to about 1000-word with that vocabulary with the read cycle of the memories 200 ns.

To alleviate the speed limitation due to huge amount of memory access, the processing elements may well be modified to be cascadable so that several consecutive trellis stages can be processed in pipeline, Since the state transition metrics depends only on the incident state and destination state, they can be fetched from the memory only at the first processing element and then conveyed to the next processing element in pipelined fashion. Also, the state metrics produced in a trellis stage don't have to be stored and refetched every frame. These curtail the memory access frequency significantly. Nonetheless since the output metrics are dependent on the input VQ index, they must be fetched every frame, thus forms a major bottleneck. The throughput will increase by the factor of the number of the casecaded stages.

Whereas above-mentioned considerations may alleviate the memory access bottleneck, these increase the pin count of chip considerably. Moreover the balance of computational speed and the I / O bandwidth are to be taken into consideration. For these kinds of problems, one of the most feasible solution is digit-serial approach[11], which will reduce the pin count and data path complexity significantly.

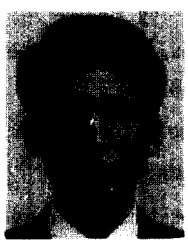
Ⅳ. CONCLUSIONS

We propose a dedicated architecture for the Viterbi scoring procedure in VQ / HMM-based real-time isolated word recognition systems. The proposed architecture can incorporate various HMMs by using tags and a flag. It can be used to recognize a word from a vocabulary of size 1 000, with the average number of states in a word HMM 50, where input VQ indices arrive every 10ms.

REFERENCES

1. L.R. Rabiner, S.E. Levinson and M.M. Sondhi, "On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition," *AT&T Bell Syst. Tech Jour.*, Vol. 62, No.4, pp. 1075-1105, April 1983.
2. L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proc. IEEE*, Vol. 77, No. 2, pp.257 286, Feb. 1989.
3. K.F. Lee, *Automatic Speech REcognition : The Development of the SPHINX System*. Kluwer Academic Publishers, Boston, 1989.
4. H. Murveit *et al.*, "A Large-Vocabulary Real Time Continuous-Speech Recognition System," *Proc. ICASSP 89*, pp. 789-792, 1989.
5. R. Bisiani, T. Anantharaman and L. Butcher, "BEAM : An Accelerator for Speech Recognition," *Proc ICASSP*, pp.782-784, 1989.
6. M.W. Koo and C.K. Un, "Compartive Study of Speaker Adaptation Methods for HMM-Based Speech REcognition," *Proc. International Conference on Spoken Language Processing*, Kobe, Japan, Nov. 1990.
7. G.D. Forney, Jr., "The Viterbi Algorithm, "*Proc IEEE*, Vol. 61, No.3, 268 279, Mar. 1973.
8. P.G. Gulak and E. Shwedyk, "VLSI STructures for Viterbi REceivers : Par I General Theory and Applications," *IEEE J. Select. Areas Commun.*, Vol. 1, No. 1, pp. 142 154, Jan. 1986.
9. P.G. Gulak and T. Kailath, "Locally Connected VLSI Architectures for the Viterbi Algorithm", *IEEE J. SElect. Areas Commun.*, Vol. 6, No.3, pp. 527-537, Apr. 1988.
10. W.G. Bliss and L.L. Scharf, "Algorithms and Architectures for Dynamic Programming on Markov Chains", *IEEE Trans Acoust., Speech, Signal Processing*, Vol. 37, No. 6, pp. 900 912, June 1989.
11. M.J. Irwin and R.M. Owens, "Digit-Pipelined Arithmetic as Illustrated By the Paste-Up System : A Tutorial", *Computer*, Vol. 20, N. 4, pp 73, April 1987.

▲Soon Young Yoon



was born in Inchon, Korea, on Dec. 10, 1965. He received the B.S. degree in elecnics engineering from Seoul National University, Seoul, in 1988, and the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology(KAIST), Seoul, in 1990. He is currently working toward the Ph.D. at KAIST. His research interests include synchronization and channel coding for high speed digital transmission systems and VLSI signal processing.

▲Hwang Soo Lee

Associate Professor, Electrical and Electronics Engineering, KAIST (Vol. 6 No. 3)