A Microprocessor based Speech Recognizer for Isolated Hindi Digits

Ashutosh Saxena and Abhishek Singh
Department of Electrical Engineering, Indian Institute of Technology Kanpur, India

ABSTRACT

A novel method for recognition of isolated spoken words on an 8-bit microprocessor is presented. The method uses a new but simple feature vector based on the zero-crossings of the speech signal. The feature vector is the histogram of the time-interval between successive zero-crossings of the speech signal. Dynamic time warping is used to calculate a time-aligned normalized distance between the feature vector and the reference templates.

The implementation needs only 1-bit A/D conversion and performs all its computations in integer arithmetic. Speaker-dependent recognition accuracy of 95.5% (*Matlab*) and 85% (microprocessor implementation) are obtained for Hindi digits spoken by 2 male speakers.

I. INTRODUCTION

Speech being a natural mode of communication for humans can provide a convenient interface to control devices. Some of the speech recognition applications require speaker-dependent isolated word recognition. Current implementations of speech recognizers have been done for personal computers and digital signal processors. However, some applications, which require a low-cost portable speech interface, cannot use a personal computer or digital signal processor based implementation on account of cost, portability and scalability. For instance, the control of a wheelchair by spoken commands or a speech interface for Simputer [1].

The implementation of a speech recognizer on a fixed-point microprocessor could provide a possible solution to such applications. Standard algorithms based on hidden markov models (HMM) and artificial neural networks (ANN) cannot be used be on a fixed-point microprocessor because these algorithms require computation which cannot be done in real-time on an 8-bit microprocessor. Hence, there is a need for a simpler algorithm.

We present a novel algorithm that uses only integer arithmetic and, hence, can be efficiently implemented on a fixed-point microprocessor in real-time. The algorithm is used for performing speaker-dependent recognition of isolated Hindi digits.

Speech recognition algorithms employ a short time feature vector to take care of the non-stationary nature of the speech signal. Standard feature vectors Mel frequency cepstrum coefficient (MFCC) or linear prediction coefficient (LPC) are computationally intensive. We designed a new but simple feature vector that uses only the zero crossings of the speech signal. The novel feature extraction requires 1-bit A/D conversion because it processes only zero crossings. The feature extraction is computationally very simple. It does not require any pre-processing of the speech signal. The feature vector preserves all information regarding the duration of the time intervals.

The short time feature vector is the histogram of time-interval between successive zero crossings of the speech utterance in a short time window. The feature vectors, extracted for each window in the speech utterance, are combined to form a feature matrix. The matrix is then normalized by multiplication with a weight vector. Dynamic time warping (DTW) [3] is used to calculate the distance between the feature matrix of the input signal and the reference patterns. DTW finds a best time-aligned path for minimum distance under certain specified constraints [3,4]. The pattern corresponding to the minimum distance is then identified to be the unknown input signal if the distance is less than a predetermined threshold.

The algorithm was implemented on *Matlab* to perform speaker dependent isolated word recognition on a vocabulary of 10 isolated Hindi digits. Simulation of the algorithm with recorded utterances gave an accuracy of 95.5%.

The algorithm was then implemented on Intel-8085, an 8-bit microprocessor. The details of the design, coding, memory and interfacing are given. With microprocessor running at a clock of 1.5 MHz, the average response time of the microprocessor is 0.9 second, with a worst-case response time of 2 seconds.

II. METHOD

The speech signal x(t) is bandpass filtered (f_L, f_H) to give s(t), which is subjected to infinite amplitude clipping to give u(t), mathematically expressed as

$$u(t) = \begin{cases} 1 & s(t) \geq 0 \\ 0 & s(t) < 0 \end{cases} \qquad (1)$$

The signal u(t) is sampled at F_S to give u[n] as shown in Fig. 1. The zero-crossing feature extraction is performed on u[n].

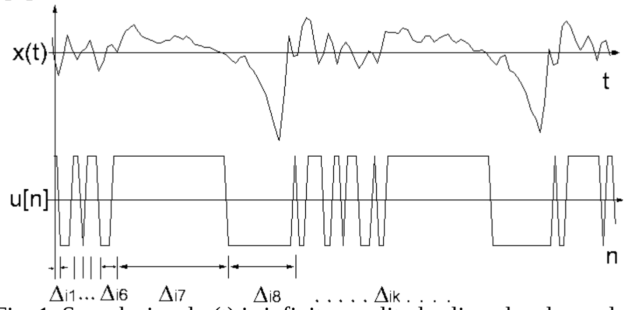

x(t)
t
u[n]
n
$\Delta_{i1} \ldots \Delta_{i6}$
Δ_{i7}
Δ_{i8}
$\ldots\ldots \Delta_{ik} \ldots\ldots$

Fig. 1. Speech signal x(t) is infinite amplitude clipped and sampled to give u[n].

1) Histogram of time-interval between successive zero crossings

The signal u[n] is windowed for every W samples without overlap. A zero-crossing analysis is carried out for each observation window. Let Z be the number of zero-crossings in the ith window. Define $\Delta_{i1}, \Delta_{i2}, \ldots, \Delta_{iZ}$ as the time-interval between successive zero-crossings in the ith window. Since the lower cut-off of bandpass filter is f_L, the highest value of Δ_{ik} can be F_S / f_L. Similarly, Δ_{ik} = 1 corresponds to the upper cut-off f_H of bandpass filter. Thus the time-intervals between successive zero-crossings can have values

$$\Delta_{ik} = 1,2,.....,M \qquad (2)$$

where

$$M = F_S / f_L \qquad (3)$$

The histogram of Δ_{ik} in the ith short time window is stored in a vector $D_i(j)$ as follows

$$D_i(j) = \sum_{k=1}^{Z} d\,[\,j - \Delta_{ik}] \qquad (4)$$

where $\delta[n]$ is the familiar dirac delta function. The length of vector D_i is M.

If T is the time duration of the isolated word, then the number of short time windows is L, where L is given as

$$L = \frac{T * Sa}{W/3} - 2 \qquad (5)$$

To obtain the histogram of the zero-crossings for the entire utterance of the isolated word we define a matrix D as follows

$$D = \{ D_1, D_2, .., D_L \} \qquad (6)$$

Fig. 2 shows the matrix **D**, which represents the histogram of time-interval between successive zero-crossings for an entire utterance. Thus, D_{ij} is the number of times the time interval between successive zero-crossings of u[n] in the ith time window is equal to j.

2) Distance Measure

A simple distance measure between the matrices does not take into account the variation and alignment in time domain. For this purpose, dynamic time warping is applied on the matrix U of the euclidean distance between the short-time feature vectors of the signal and the reference pattern.

$$U_{xy} = \sum_{j=1}^{M} [D_{xj} - T_{yj}]^2 \qquad (7)$$

where

T	matrix of size L_R x M, of the reference pattern of length L_R.
D_x	vector in xth time window of the input signal.
T_y	vector in yth time window of the refernce pattern.
U_{xy}	the distance between the vectors D_x and T_y

3) Dynamic Time Warping

Different acoustic renditions of the same speech utterance are seldom realized at the same speaking rate across the entire utterance. Hence, when comparing different renditions of the same utterance, speaking rate variation as well as duration variation should not contribute to the dissimilarity (distance).

Time alignment in DTW is done along a best-aligned path which has the least accumulated distance [4]. Certain constraints are placed on the path to make time alignment meaningful.

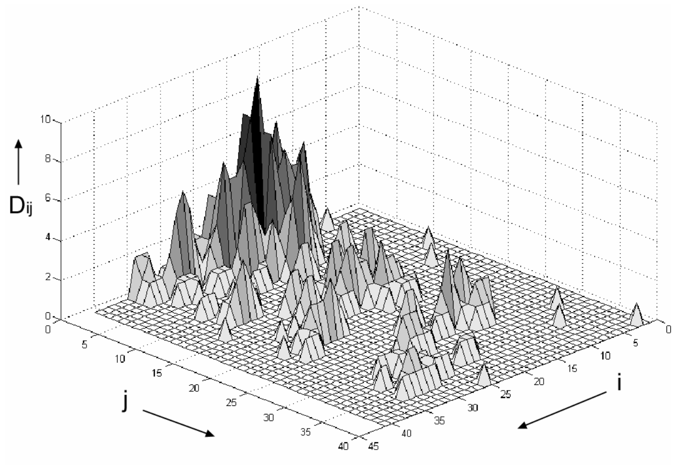


Fig 2. Surface plot of matrix D for a complete isolated utterance by a male speaker.

A. Endpoint Constraints

Different renditions of the isolated utterances usually have well defined endpoints that mark the beginning and the ending frames of the pattern. For time normalization the end points are the fixed temporal limits of the utterances. To take into account inaccuracy in the end point detection, the end point constraints are relaxed by one window corresponding to a 16 ms region. This represents the maximum anticipated uncertainty in the endpoints of the input signal.

B. Local Continuity Constraints and Slope Weighting

To ensure proper time alignment while keeping any potential loss of information to a minimum local continuity constraints on the path were incorporated as shown in Fig. 3. The local continuity constraints and slope weights are empirically determined. The local constraints are expressed in terms of allowable paths as

$P_a \rightarrow (1,1)(1,0)$

$P_b \rightarrow (1,1)$

$P_c \rightarrow (1,1)(0,1)$

All the allowed local paths are equally weighted.

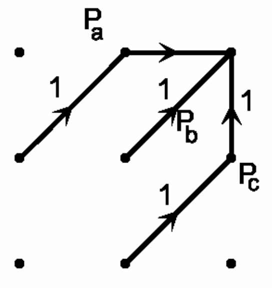


Fig. 3. Local path constraints with slope-weights.

C.Global path Constraints

This additional global path constraint precludes any path that involves excessive time stretch or compression and effectively reduces the area of computation. An additional global constraint as proposed by [3] is used. T_0 is the maximum allowable time deviation between the two patterns at any window. T_0 is taken to be $(L+L_R)/6$.

These constraints are applied to find the minimum accumulated distance in the matrix U. The scalar distance v(U) of matrix U is the minimum accumulated distance along paths allowed by the constraints. To make the distance independent of the lengths of the two patterns being compared it is divided by a normalizing factor equal to the sum of the lengths of the two patterns.

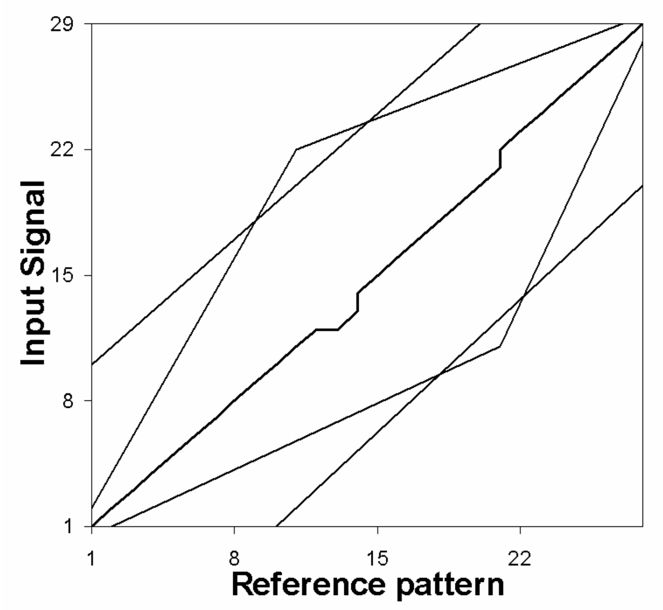


Fig. 4. The constraints used in Dynamic Time warping, shown along with the path taken.

4) Classification

The scalar distance v(U) between the input signal and each of the reference patterns is calculated. The minimum of these distances is found. If this minimum is below a certain predetermined threshold then the corresponding reference pattern is identified as the input signal; otherwise no pattern is identified. This prevents the algorithm from recognizing any arbitrary noise signals.

III. SOFTWARE DESIGN

The code for realtime feature extraction is shown in Fig 5. The infinite amplitude clipped speech signal is received by serial input pin and stored temporarily in A. The current sample (in A) is compared with the previous sample (in B) to determine if a zero-crossing has occured. The counter C counts the number of samples since the previous zero crossing. If a zero-crossing has occured then the entry of the feature matrix corresponding to the current time window (stored in E) and the zero-crossing interval (stored in C) is incremented. We keep f_L = 125 Hz, so the maximum time interval between successive zero crossings is 32. The lower 5-bits of the address of the matrix are, therefore, calculated by counter C; and the higher bits by window number E. Thus, the time interval between successive zero-crossings is calculated and updated in the feature vector of the current window. Counter D keeps track of the total number of samples, and is used to detect end of a window.

Fig. 6 shows the flowchart for the rest of the algorithm. Distance calculation between the input feature matrix and the pre-stored matrix of feature pattern is performed by dynamic time warping (DTW). The euclidean distance is calculated by using a look-up table, which stores the squares of integers. An efficient subroutine is coded which returns the minimum of distances for the local paths P_1, P_2, P_3.

Classification algorithm is a simple routine for finding the minimum of the calculated distances for each utterances. If the minimum distance is above a predetermined threshold then the utterance is discarded and no utterance is identified. While calculating distance with a particular utterance, if the distance is very small (i.e. smaller than another threshold) then that utterance is identifed immediately and no further comparisons with other utterances are made to decrease response time.

The identified reference pattern is sent out using serial output. The 4-bit serial output representation of the identified utterance is sent through SOD.

In “training” mode, the feature matrices formed for each utterance are stored in EEPROM.

Table 1: Memory Organization

Type of memory	Name	Size
ROM	Code	1 KB
	Lookup Table	50 bytes
	Templates	6 KB
RAM	Feature matrix D	1KB
	Distance matrix U	1KB
	Scratch Pad	256 bytes
	Stack	256 bytes

IV. HARDWARE DESCRIPTION

Fig. 7 shows the block diagram of the hardware. The input voltage signal is bandpass filtered and hardlimited (1-bit quantized) by using a comparator. The quantized output of the comparator is fed into the Serial Input (SID) of the 8085A. An active local peak detector circuit is made using a Opamp to detect end points of the utterances. The output of the Active Local Peak Detector followed by a comparator acts as an interrupt for the 8085A to indicate the starting of the utterance. It will go low when the utterance ends.

The system consists of the microprocessor 8085A, EEPROM (8k), RAM (8k), and a register to demultiplex and latch address and data lines. A decoder was used for the chip select of memory chips. The microprocessor was run at a clock speed of 1.5 MHz.

Another interrupt is used for indicating the mode of operation. The circuit in "training" stores the reference patterns. If this interrupt is given 8085A goes into "training" mode and stores the matrix of the input signal as reference permanently in the EEPROM.

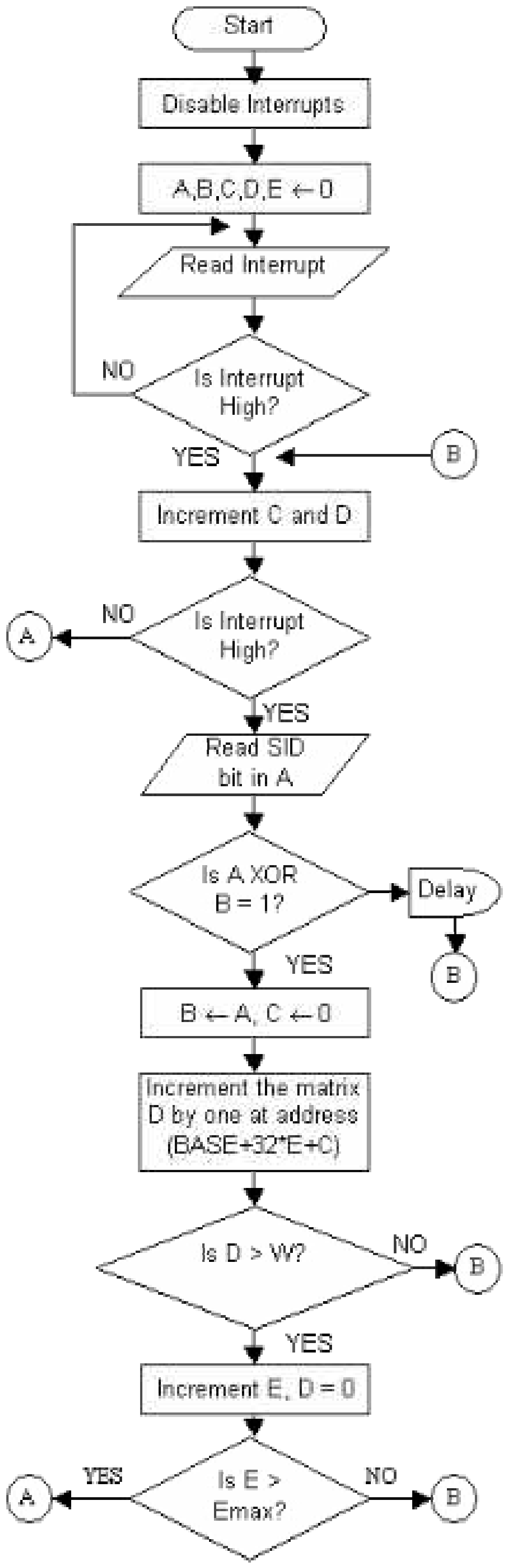
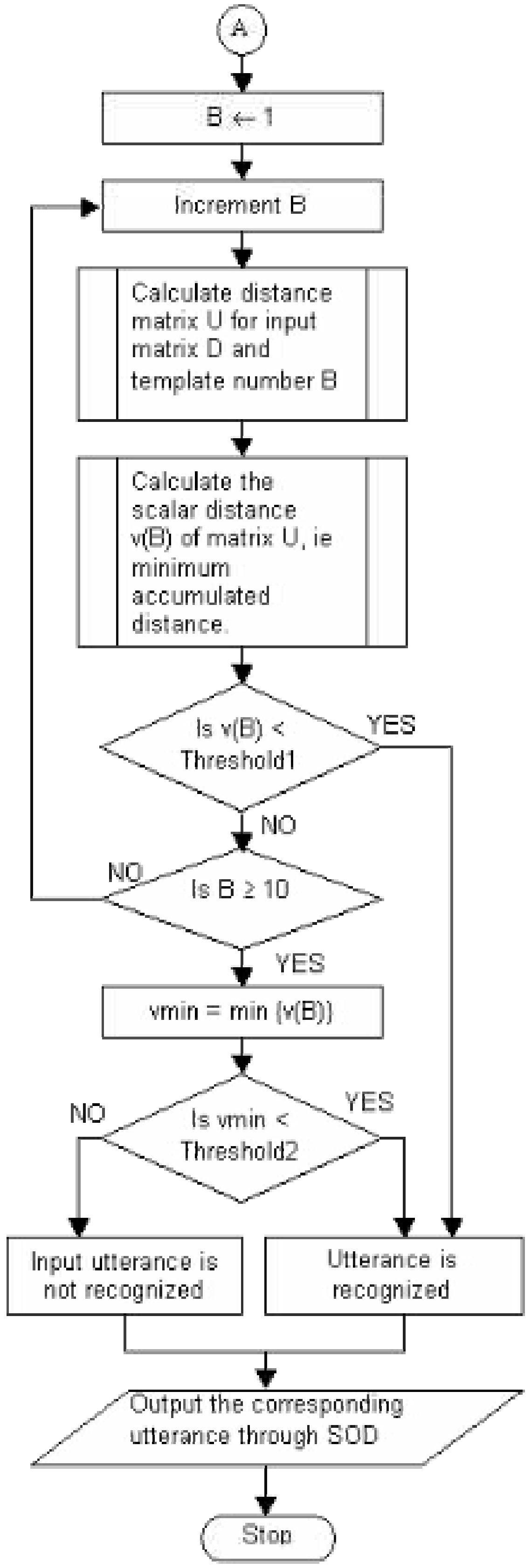


Fig 5. Flowchart for real-time code for feature extraction.



A
B ← 1
Increment B
Calculate distance matrix U for input matrix D and template number B
Calculate the scalar distance v(B) of matrix U, ie minimum accumulated distance.
Is v(B) < Threshold1
YES
NO
NO
Is B ≥ 10
YES
vmin = min (v(B))
NO
Is vmin < Threshold2
YES
Input utterance is not recognized
Utterance is recognized
Output the corresponding utterance through SOD
Stop

Fig 6. Flowchart for distance calculation and classification part.

V. RESULTS

The algorithm was simulated on *Matlab*. The sampling frequency F_S used was 8kHz. The cutoff frequencies of the bandpass filter were f_L = 100 Hz, to remove the 50 Hz ac noise, f_H = 3.4 kHz to prevent aliasing. The length of the non-overlapping window W was 128 samples, which corresponds to a 16 ms region. Four renditions of each digit were used to make the references patterns. The results for simulation of the algorithm on Matlab are given in Table 2.

For hardware implementation on the microprocessor, f_L was 125 and f_H = 3.4 kHz, with a sampling frequency of 8 kHz. The results are given in Table 2. With microprocessor running at a clock of 1.5 MHz, the average response time of the microprocessor is 0.9 second, with a worst-case response time of 2 seconds.

Table 2: Results for Hindi digits for two speakers

S. N.	Utterance	Size of test data	Accuracy (%)			
			Matlab		Microprocessor	
			1	2	1	2
1.	*Ek*	100	99	99	82	86
2.	*Do*	100	97	98	91	87
3.	*Teen*	100	98	96	87	89
4.	*Chaar*	100	97	93	87	83
5.	*Paanch*	100	94	95	83	79
6.	*Chhah*	100	89	86	78	82
7.	*Saat*	100	95	96	85	86
8.	*Aath*	100	93	91	83	81
9.	*Nau*	100	99	99	92	88
10.	*Shoonya*	100	99	98	87	90
Total			96.0 %	95.1 %	85.5%	85.1%
			95.5%		85.3%	

VI. CONCLUSION

An algorithm for performing speaker-dependent isolated word recognition on an 8-bit microprocessor is presented. The main contribution of the work is the use of a new but simple feature vector based on histogram of time interval between successive zero crossings of the speech signal. The development of a microprocessor based speech recognizer is reported. This can be used as an embedded system for control tasks in devices such as a wheel chair and Simputer.

REFERENCES

[1] The Simputer Trust. Simputer™: What is a Simputer? [Online] Available: http://www.simputer.org/simputer/about

[2] Lipovac and V. Sarajevo, "Zero-crossing-based linear prediction for speech recognition", *Electronics Letters*, pages 9092, vol. 25 Issue 2,19 Jan 1989.

[3] H. Sakoe and S. Chiba, "Dynamic Programming Optimization for Spoken Word Recognition", *IEEE Trans. Acoustics, Speech, Signal Proc.*, ASSP-26(1):43-49, February 1978.

[4] Lawrence Rabiner, and Biing-Hwang Juang, "Fundamentals of Speech Recognition", *PTR Prentice Hall, Englewood Cliffs, New Jersey 07632*, 1993.

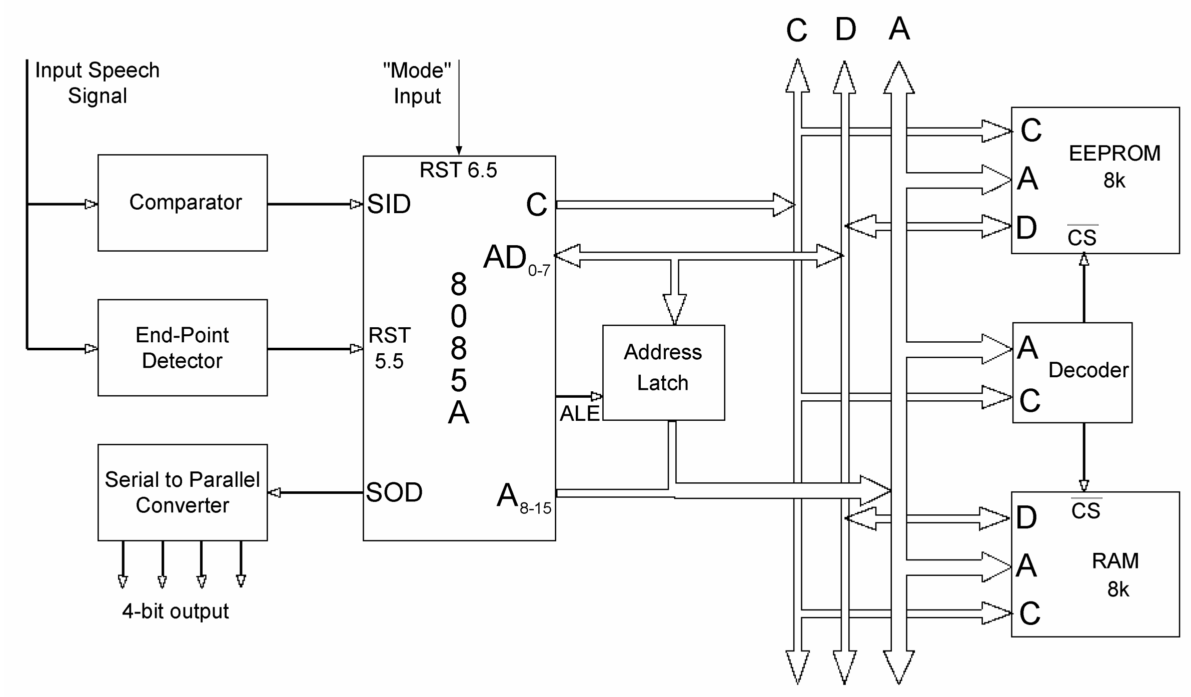


Fig. 7. The hardware block diagram