AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com

FPGA-Based Implementation of Text Analyser and Syllable Preparation for Concatenative Speech Synthesis of Tamil Language

[1]T. Jayasankar and [2]Dr.J.Arputha Vijayaselvi

[1]*Assistant Professor, Department of Electronics and Communication Engineering, Anna University BIT Campus, Tiruchirappalli,Tamilnadu*
[2]*Dean (R&D) Department of Electronics and Communication Engineering, Kings College of Engineering, Pudukkottai, Tamilnadu*

ARTICLE INFO

Article history:
Received 25 April 2014
Received in revised form
8 May 2014
Accepted 20 May 2014
Available online 17 June 2014

Keywords:
Concatenation Text to Speech Synthesis (CTTS), Text analysis, Tamil Language, FPGA.

ABSTRACT

This paper describes about the design and development of an FPGA based Text Analysis and syllable preparation for Concatenation Text to Speech Synthesis (CTTS) of Tamil Language. Research on TTS conversion is a large enterprise that shows an impressive improvement in the last couple of decades. A complete implementation of CTTS is composed of Text analysis, prosody control, and speech synthesis. A text analysis is the TTS front end deals with converting text in to pronunciation form, called Text Normalization and get the prosodic information that may be obtainable by analysing various underlying structures of the text. The texts are converted to the pronunciation form using text analysis module in natural speech synthesis system. Then development of CTTS is carried out by addressing the text analysis issues include text normalization, numerical words, abbreviation and syllable segmentation w The implementation of this system will be tested using Xilinx FPGA VIRTEX V Board. This hardware implementation through FPGA makes the text analyser eases the integration of any other embedded devices for TTS.

To Cite This Article: T. Jayasankar, Dr.J.Arputha Vijayaselvi., FPGA-Based Implementation of Text Analyser and Syllable Preparation for Concatenative Speech Synthesis of Tamil Language ***Aust. J. Basic & Appl. Sci.,*** *8(10): 102-109, 2014*

INTRODUCTION

In recent years, Text to speech synthesis has become one of the hottest technologies that the world powers desperate to focus on. It has integrated language and speech for human computer interaction. Text to Speech (TTS) Synthesis is an automated encoding process which converts a sequence of symbols, (text) conveying linguistic information, in to an acoustic waveform (speech), whether it was directly introduced in the computer by an operator or scanned and submitted to an Optical Character Recognition (OCR) system. Getting output is simple for user but the process of converting text into speech is not an easy task. It involves linguistics, phonetics and speech signal processing technology of digital coding, and it is necessary to synthesize high-resolution, naturalness conformity with specific language environment of speech flow.

A complete implementation of CTTS is composed of Text analysis, prosody and speech synthesis. G.L.Jayavardhana *et al.*,(2001). A text analysis is the TTS front end deals with converting text in to pronunciation form, called Text Normalization and analyzing various underlying structures of the text. The necessary information from the front end given to the back end of synthesis system. As seen in the simplified block diagram of Concatenative Text to speech synthesis system in Fig. 1. The analysis of text processing is an important component in language and speech technology applications such as TTS ,Multi lingual language,and Speech recognition system Saraswathi(2010). The text analysis development can be carried out specifically for each language. Here our attempt is to describe the methodology used for text analysis of Tamil language.

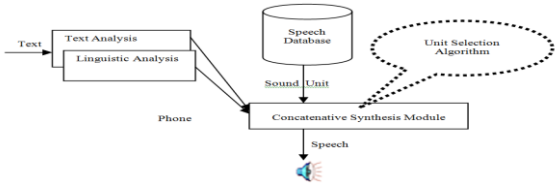


Fig. 1: Block diagram of Concatenative Text to speech synthesis system.

Corresponding Author: T. Jayasankar, Assistant Professor, Department of Electronics and Communication Engineering,Anna University BIT Campus, Tiruchirappalli,Tamilnadu,India
E-mail: jayasankar_t@rediffmail.com

Speech synthesis system are classified into G.L.Jayavardhana *et al.*,(2001) three broad classes (1) Articulatory synthesis (2) Parametric synthesis (3) Concatenative synthesis. Each of this system has its own strengths and weaknesses and the intended users of a synthesis system will typically determine which approach will be used. In these three methods the Concatenative approach seems to be the industry's current favorite as it can offer the most natural sounding speech. The Concatenative speech synthesizer produces speech by concatenating the small, pre recorded units of speech data base. In Concatenative speech synthesis system, the speech signal is segmented into some basic sound units like words, phonemes, syllables. For building Indian language speech synthesis systems, the syllabification unit should be efficient S. Rao, (2005), M.Samue(2006), Narendra(2011), A Lakshmi (2006). Some of the advantages of using syllables as basic unit is, they have fairly long duration when compared to phonemes or di-phones. Hence, the task of segmentation becomes relatively easier. Also, since the boundaries of most of the syllables are low energy regions (due to consonants), the concatenations would result in reduced perceivable distortions.

Various researches have been done to synthesize speech by different means and for different languages as shown Table 1. In Indian Language Dhvani TTS system [15], they use phoneme concatenation technology with an attempt to cover use phoneme concatenation technology with an attempt to cover all Indian languages under a single framework. This system can detect the languages and it dispatches the text to the corresponding phonetic synthesizer. Considerable work may has been done in text normalization of other language English, Bangala, Hindi. Many little works has been done on Tamil language.

Table 1: TTS system.

Researcher	Language	Synthesis	Unit/Database
TIFR Mumbai	Hindi,Bangali,Marathi	Formant Synthesis	Phonemes
CEERI Delhi	Hindi,Bangali	Formant Synthesis	Phonemes & syllable
IIT Chennai	Hindi,Tamil	Diphone synthesis	Syllable
CDAC Noida	Hindi	Concatenative	Syllable and words
CDAC Kolkatta	Bangali	Concatenative	Phoneme and sub phonemes

The basic model Lu Gao *et al.*,(2010), K. Panchapagesan *et al.*,(2004) of a text analysis system is the same for all languages except a language dependent rules and framework. When the text analysis is processed with the proposed CTTS method, the system is found to produce intelligible and more natural sounding speech. So far, the framework for Indian languages has been done using software. The increase the speed of the front end and to provide cost effective solution.

The previous approaches for Tamil TTS and other language text analysis system were mostly developed using festival framework or C++ M.Samuel *et al* (2006), G.L.Jayavardhana *et al.*,(2001). The software based approach requires a high performance PC with a relatively large database, whereas the power consumption and physical size is more. In this system sequential mode of operation is carried out. The performance in PC based system is relatively low when the actual throughput is high in real-time. However, when mass text documents need to be translated, a faster method for text normalization is obviously preferred. Although software implementations of text analysis and speech synthesis systems are capable of real time operating, growing number of mobile devices implies the need for hardware implementation of speech synthesis systems.

The performance of the text analysis system of Tamil Language is significantly improved using a pipelined parallel system using FPGA and is the main contribution of this work. To improve the performance of the system, FPGA offers improved flexibility, better performance and relatively less power. The hardware approach paves a way to speed-up the searching of text and speech. This paper highlights the implementation of text analysis in a Xilinx FPGA evaluation platform to process large text inputs into syllabified output pattern in fast manner for TTS.

Introduction to Tamil Language:

Tamil belongs to the Dravidian language family. This classical language is considered the earliest of the Dravidian languages and is spoken by more than eighty million people worldwide. Tamil is regarded as one of the four major literary languages of the Dravidian family and, in spoken form, is predominant form of communication in Tamil Nadu in south India. There are thirty characters in the Tamil ethnographic system. There are twelve vowels and eighteen consonants. The basic units of the writing system in Indian languages are characters which are an orthographic representation of speech sounds T.Jayasankar (2010). A character in Indian language scripts is close to a syllable and can be typically formed by a vowel or by a consonant following

a vowel. There is fairly good correspondence between what is written and what is spoken. The Tamil alphabet is syllabic, in that each letter denotes a syllable.

Architecture of Textual Analysis:

As most of the speech synthesis system converts text into speech, here we have considered text as input for this work. The main requirement is that the text must be in Tamil language or in Transliteration format. In our case, the input text is in manual entry. The main aim of the text analysis is to process the given input text and produce the actual pronounceable representation from the written text. Textual analysis system comprises these four fundamental components as shown Fig 2.

1. Text Normalization
2. Parser Sequence
3. Word to syllable converter
4. Dictionary rules

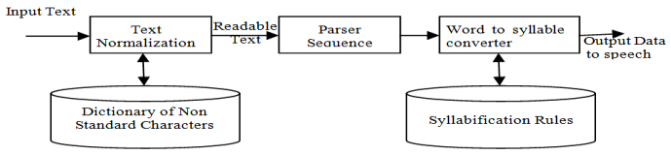


Fig. 3: Block diagram Textual analysis

Text Normalization:

The input text is read either in word, sentence and paragraph format. Normally input to a TTS system is not always pure text and it may contains grammatically composed sentences, and other special characters such as numbers, quotes, time, date ,abbreviations etc that have separate syntax which need to be first converted to the own language graphemic form. The identification of each syntax and translating them in to pronounceable word form is called text normalization. The text normalization is the first step for Tamil text to speech system. The Text normalization plays a vital role in text analysis and its effect has direct impact on the correctness of the text in the speech information. It subsumes sentence segmentation, tokenization and normalization of non standard words.

Sentence Splitting:

The first task in text normalization is sentence splitting. In order to segment the given Tamil sentence in to separate word for speech synthesis; we need to know that end point of the sentence. Most of the sentences will be ending with full stop, but some other cases where it ends with semicolon or some other punctuation (! ?). The sentence in tokenized end of the word in (.) is easily identified. Therefore sentence boundaries can be determined, dividing the text into tokens separated by whitespace. The main issue is the indistinctness of whether it marks sentence boundary, Number digit or abbreviation. There comes the abbreviation, for this there is full stop coming in between the sentence for the abbreviation. Therefore solve this issue by expanding the abbreviation that will replace the text in normal form using database approach. There could be other special character such as quotes, comma, semicolon etc it must be removed from the sentence before normalization.

Abbreviation:

If that abbreviation is present in the sentence, it will replace full form of the text while reading. In case of an abbreviation, normalized representation can be obtained from the database, if not it will be leaving the original text as it is. Database consists of some mostly used abbreviations collected from various newspapers, books and some of the examples are shown in Table 2.

Table 2: Abbreviation.

Abbreviation	Expanded in Tamil
ரூ.	ரூபாய்
த.நா	தமிழ் நாடு
மி.மீ	மில்லி மீட்டர்
எ. கா	எடுத்து காட்டு
செப்.31	செப்டம்பர் 31

Normalization of Non Standard Words-Number Converter:

In text normalization another important issue is normalizing non-standard words like number converter which need to be expanded into sequences of Tamil words before they are pronounced. Number expansion is whether the encountered number is an ordinary number or phone number or date or time etc. Numbers produce

varying pronunciation depending on the way of treatments. If the type is identified, the number can be normalized based on the rules. For example if a sentence has a number ending with special symbol character such as (/,-) which can be treated as date. In the pronunciation module the number is pronounced, based on the dictionary rules as shown in Table 3.

Sentence with Number	Treated as	Pronunciation in Tamil	Sentence Analysis Information
04/10/1980 04-10-1980	Date of birth	நாண்கு பத்து ஆயிரத்து தொளாயிரத்து எண்பது	This sentence has three numeric words separated with /, - which indicates as a date.
RU .10	Money	ரூபாய் பத்து	This sentence has a number started with ru which indicates as money. It should not be pronounced as isolation.
9994447711	Phone Number	ஒன்பது ஒன்பது ஒன்பது நாண்கு நாண்கு நாண்கு ஏழு ஏழு ஒன்று ஒன்று	The number string length is equal to ten 10 which indicates as a phone number..Respective pronunciation is decided as individual character not be as whole.
Vandy en. 4471	Train number	வண்டி எண் நாண்கு நாண்கு ஏழு ஒன்று	The no started with . Which indicates the pronounced as individual number.
10980	Total value /volume	பத்தாயிரத்து தொளாயிரத்து எண்பது	In this case number is expanded based string length Example 4 character length 4th position in terms of ayiram and 3 rd position in terms of nooru etc.

Syllable Parser Unit:

A syllable parser unit consists of two modules. They are (1) parser sequence and (2) Word to syllable converter block. In Parser Sequence block, the input receives from the Text- normalization unit is a string of characters and it converts the string into consonant and vowel symbol.

Example அப்பா அ+ப் + ப்+ஆ VCCV பறவை ப் +அ+ ற்+அ+வ்+ஐ CVCVCV

Word to Syllable Converter:

Building of our Tamil TTS, syllables are used as synthesis unit. The general structure form the Indian language syllable is C*VC*, where C is a consonant, V is a vowel and C* indicates the presence of 0 or more consonants. A Syllable Parser functioned to extract the Tamil Syllables from parser sequence unit and arranged it according to the sequence of syllables based on Tamil Linguistic rules. According to Tamil Text synthesis system the syllable structures are V, VC, VCC, CV, CVC, CV, and CVCC.

Rules for Segmentation of Tamil:

Some of the rules used to perform text to syllable conversion [4], [7] are:

- Nucleus can be Vowel (V) or Consonant (C).
- If onset is C then nucleus is V to yield a syllable of type CV.
- If the characters after CV pattern are of type CV then the syllables are split as CV and CV.
- Coda can be empty of C.
- If the CV pattern is followed by CCV, then syllables are split as CVC and CV.
- If a CV pattern is followed by CCCV, then the syllables are split as CVCC and CV.
- If the VC pattern is followed the V, then the syllables are split as V and CV.
- If the VC pattern is followed by CVC, then the syllables are split as VC and CVC.
- If the characters after CV pattern are of type CVCV then the syllables are split as CVCVCV.

Proposed Hardware Architecture Of Text Analysis:

The system architecture used to implement the complete Text analysis on algorithm has been developed using the Xilinx XC5VLX50T FPGA. The system is composed, of several devices are,

- Programmable clocks up to 400MHz.
- External 5V 4A or greater power source with a coax center-positive 2.1mm internal-diameter plug and real-time power monitors on all power rails.
- Multiple USB2 ports for programming, data, and hosting and one PS2 port to accommodate keyboard or mouse.
- 256Mbyte DDR2 SODIMM with 64- bit wide data.
- 16Mbyte Strata Flash™ for configuration and data storage 256Mbit Mnemonics P30 parallel flash memory device (organized as 16-bit by 16Mbytes) for non-volatile storage of FPGA configuration files.

- Configuration files are stored using the byte-peripheral interface mode (BPI) in either up or down configurations.
- A single FPGA configuration file requires less than 16Mbits, leaving 140Mbits available for storing the users data.
- Audio Codec with line-in, line-out, mic, and headphone. Audio data at up to 18 bits and 48-kHz sampling is supported, and the audio in (record) and audio out (playback) sampling rates can be different.
- Two 68-pin VHDC connectors for high- speed parallel I/O and four 8-pin Pmod connectors for lower speed and lower pin-count I/O. It is responsible to a large collection of accessory boards that can attach to the Pmod and VHDC expansion connectors to add ready-made functions like A/D's, D/A's, TFT Display and other functions.

Fig. 3 shows the block diagram of this architecture. The FPGA is the main component that consists of scan code extraction block which extract the scan code from the input text and text normalization block converts all the input in to readable form. The word to syllable block convert all word into syllable form. The Flash memory unit is employed to store the program and stores all the database rules. The syllabified output is forwarded to speech synthesizer unit and put in display module. The flow chart of complete text analysis system as shown Fig4.

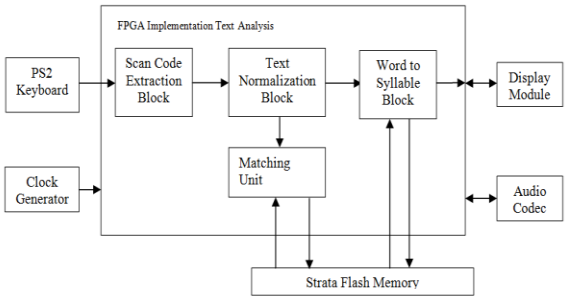


Fig. 3: System Architecture for Text analysis system.

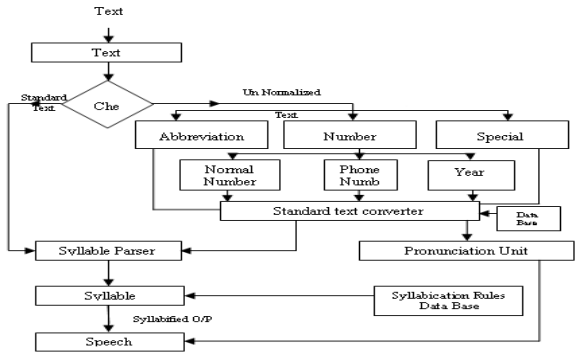


Fig. 4: Flow chart of Text analysis system.

Ps2 Keyboard:

The PS2 keyboard are used for given the input text to the FPGA which further convert the scan code from the PS2 to the equivalent ASCII. There are two different types of scan codes - make codes and break codes.

(i)Make code - A make code is sent whenever a key is pressed or held down.

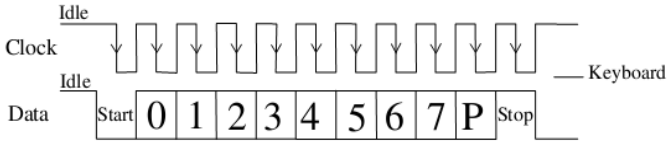
(ii)Break code- A break code is sent when a key is released. The break code is the make code preceded by 'F0'h byte. The keyboard sends packets of data, scan codes, to the host indicating which key has been pressed. The Vowels and Consonants of Tamil along with transliteration scheme as shown in Table 4.

Scan Code Extraction:

The keyboard transmission protocol is a serial format, with one line providing the data and the other line providing the clock. The data length is 11 bits with one start bit (logic 0), 8 data bits (LSB first), odd parity bit and a stop bit (logic 1). The clock rate is approximately 10 to 30 kHz and varies from keyboard to keyboard. Synchronize the external inputs to the system clock. Creates an internal copy of the keyboard signals that only changes values on the negative edge of the ps2 clock. Every negative edge of kb_clock data is fetched. After extraction of the scan code convert into Hex code for the purpose of assigning the character

Table 4: Vowels and Consonants of Tamil Along with Transliteration Scheme.

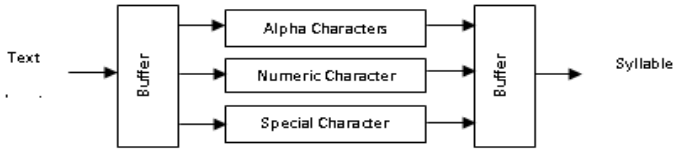
CONSONANTS				VOWEL			
க்	k	ம்	m	அ	a	ஓ	o
ங்	G	ய்	y	ஆ	A	ஓ	O
ஞ்	J	ர்	r	இ	i	ஔ	au
ச்	c	ல்	l	ஈ	I	ஃ	H
ட்	d	வ்	v	உ	u		
ண்	N	ழ்	z	ஊ	U		
த்	T	ள்	L	எ	e		
ந்	j	ற்	R	ஏ	E		
ப்	p	ன்	n	ஐ	ai		



The inputs and outputs of the circuit as defined are as follows:
kb_data -The scan code bits enters through this input.
kb_clk - The keyboard clock signal enters through this input.
sys_clk - The system clock used to clock the system.

Text Normalisation:

The scan code output pass through next important block is called text normalization. Before text normalization the retrieval of abbreviation and acronyms is crucial for appropriate sentence segmentation and normalisation of non standard words, the identification task is carried out prior to text normalization. Based on



the tokenization, the input string splits into various character blocks and then converted into readable form of text based on rules database. The readable texts collected through the buffer block are passed into syllable parser.

Word to Syllable Converter:

The word to syllable block convert all word into syllable form from the parser unit. Then the text is syllabified based on linguistic rules discussed in Sect.3. The syllabified output of sample text is given below

Avar (அவர்) அ + வ் +அ+ர் (VCVC) a var (V / CVC)

Malar (மலர்) ம்+ அ +ல்+ அ +ர் (CVCVC) Ma lar (CV/CVC)

The output of normalization unit then passes to segmentation unit. The segmentation unit gets the data and call the suitable data char from memory. The memory unit only has the data base of Tamil characters which is in the form of audio codec input. The segmentation unit may get the data and generate the delay for audio control and then send the data char to audio codec to generate speech output.

Experimental Result:

The text analyser system is tested on two different databases from newspaper, commonly used words and story books, also different domain such as news, education and sports containing 1500 words in total. In database1 consists of 1000 standard words are split into syllables are accurately 99%.In database 2 contains both standard and non standard words of 500 are normalaised accurately 88%.The result is shown is Fig 4.

Implementation of Tamil text analyser is done in VIRTEX V target hardware as shown in Fig 5. This system only requires low memory and the number of words to be processed is lead to infinite. Table 4 shows the Design Summary of the text analyser. So these systems provide a way for a robust standalone, high speed Text to Speech synthesis.

This system only requires low memory and the number of words to be processed is lead to infinite. Table 4 shows the Design Summary of the text analyser. So these systems provide a way for a robust standalone, high speed Text to Speech synthesis.

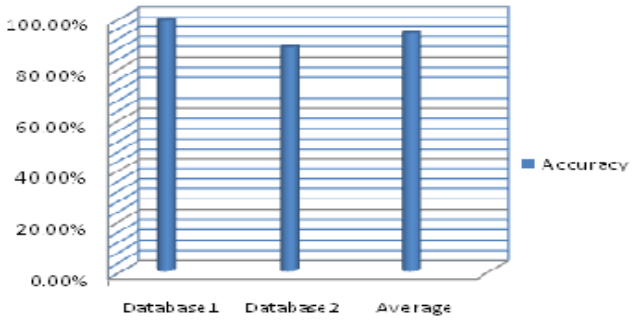


Fig. 4: Accuracy of the Text Analyser System.

Fig. 5: FPGA Implementation of the text analyser design.

Table 4: Design summary of the text analyser.

lcd_key Project Status			
Project File:	lcd_key.ise	**Current State:**	Programming File Generated
Module Name:	ps2_lcd_new	**• Errors:**	No Errors
Target Device:	xc5vlx50t-1ff1136	**• Warnings:**	49 Warnings
Product Version:	ISE 10.1 - Foundation	**• Routing Results:**	All Signals Completely Routed
Design Goal:	Balanced	**• Timing Constraints:**	All Constraints Met
Design Strategy:	Xilinx Default (unlocked)	**• Final Timing Score:**	0 (Timing Report)

lcd_key Partition Summary
No partition information was found.

Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**N**
Number of Slice Registers	211	28,800	1%	
Number used as Flip Flops	204			
Number used as Latches	7			
Number of Slice LUTs	391	28,800	1%	
Number used as logic	378	28,800	1%	
Number using O6 output only	341			
Number using O5 output only	31			
Number using O5 and O6	6			
Number used as Memory	11	7,680	1%	

Conclusion:

This paper shows that the proposed Text analysis system design is an effective and easy method of produce an pronunciation form corresponding to an arbitrary text input . This system concentrated on syllabification unit which used for preprocessing the input text database and gives the syllabified output data array for synthesizer. The syllabification unit which is designed and implemented on Xilinx FPGA VIRTEX V board also pushed for test bed. Furthermore, the FPGA implementation allows the usage in hardware platforms where low power consumption is required. Here we reduce area consumption and also power consumption by reusing same hardware for various stage of computations The system known as very effective for standalone application and can be implemented with lesser complexity. The advantage of this mechanism is that it will improve the synthesizer quality much greater for the naturalness and understandability of the speech. The realization on an FPGA makes us to believe that such text analyser can be incorporable with any other embedded devices of TTS. The future work text analysis module integrated in to back end of concatenation of the Tamil speech synthesis into a single chip manner and also to produce more natural output speech in effective manner than software TTS. This work is the first successful try for Text analyser in CTTS for Tamil language.

REFERENCES

Aniruddha Sen, 2007. Speech Synthesis in India IETE Technical Review, 24(5): 343-350.
Http://dhvani.sourgeforge.net, (Dhvani-TTS System for Indian Languages), 2001.

Jayavardhana, G.L., Rama, A.G. Ramakrishnan, R. Muralishankar and Vijay Venkatesh, 2001. Thirukkural – A text to speech synthesis system, Proc. Tamil Internet 2001, Kuala Lumpur, 92-97.

Jayasankar,T., R. Thangarajan, J. Arputha Vijaya Selvi, 2011. Automatic Continuous Speech Segmentation to Improve Tamil Text-to-Speech Synthesis, International Journal of Computer Applications, 25-1.

Jayasankar, T., J. Arputha vijaya selvi, R.Rajendran, 2010. Realization of Tamil syllables Text to Speech Transferring System using FPGA, in Tamil Internet Conference Coimbatore, 23-27.

Jayasankar, T., J. Arputha Vijaya Selvi, 2011. Embedded Unit Selection with Limited Data Base for Tamil Text-to-Speech Synthesis , *CiiT* International Journal of Programmable Device Circuits and Systems.

Kishore, S.P. & A. Black, 2003. Unit size in unit selection speech synthesis. In Proc.of EUROSPEECH, 1317–1320.

Lakshmi, A. and Hema, A. Murthy, 2006. A Syllable-based Continuous Speech Recogniser for Tamil, INTERSPEECH, Pittsburgh, 1878-1881.

Lu Gao and Hongzhi Yu, 2010. *A* Research on Text Analysis in Tibetan Speech Synthesis, Proceedings of the 2010 IEEE International Conference on Information and Automation, 20-23. Harbin, China.

Muhammad Mas ud Rashid, Md . Akter Hussain, 2010. Text Normalization and Diphone Preparation for Bangla Speech Synthesis, Journal of Multimedia, 5(6): 551-559.

Narendra, N.P., K.S. Rao, K. Ghosh, R.R. Vempada and S. Maity, 2011, Development of syllable-based text to speech synthesis system in Bengali,Int. J. Speech Technol, 14(3): 167–181.

Panchapagesan, K., Partha Pratim Talukdar, N. Sridhar Krishna, Kalika Bali, A.G. Ramakrishnan, 2004. Hindi Text Normalization, Fifth International Conference on Knowledge Based Computer Systems (KBCS), Hyderabad, India, :19-22.

Rao, S.M.N. S. Thomas, T. Nagarajan & H.A. Murthy, 2005. Text-to-speech synthesis using syllable like units. In Proc. of national conference on communication (NCC), IIT Kharagpur, India, 227–280.

Ramakrishnan, A.G., Lakshmish, N. Kaushik and M. Laxmi Narayana, 2007. Natural Language Processing for Tamil TTS, Proc. 3rd Language and Technology Conference, Poznan, Poland, 5-7: 192-196.

Samuel, M., Thomas, M. Nageshwara Rao, Hema A. Murthy and C.S. Ramalingam, 2006. Natural sounding TTS based on syllable-like units, in the proceedings of the 14th European Signal Processing Conference, Florence, Italy.

Saraswathi, S., R. Vishalaksh, 2010. Design of Multilingual Speech Synthesis System, International Journal of Intelligent Information Management, (2): 58-64.

Venugopalakrishna, Y.R., M.V. Vinodh, H.A. Murthy & C.S. Rama-lingam, 2008. Methods for improving the quality of syllable based speech synthesis. In Proc. of spoken language technol-ogy (SLT) workshop, . Goa : 29–32.