International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.3, 2015

REVIEW ARTICLE

ISSN: 2321-7758

TEXT TO SPEECH CONVERSION ON INTEL ATOM PROCESSOR

S.M.K.CHAITANYA[1], N.SRUTHI[2]

[1] Assistant Professor, [2] M.Tech. Student

Department of ECE, GayatriVidyaParishad College of Engineering, Visakhapatnam, Andhra Pradesh, India

Article Received:17/05/2015 | Article Revised on:28/05/2015 | Article Accepted on:08/06/2015

ABSTRACT

The real time hardware implementation of Text-to-Speech system has been gaining its importance in the recent years due to its various real time applications. These include reading aids for the blind, talking aid for the vocally handicapped and training aids and other commercial applications. The methodology used in TTS is to exploit acoustic representations of speech for synthesis together with linguistic analyses of text to extract correct pronunciations and prosody in context. In this, the embedded processor IFC946X Intel atom processor has been chosen as hardware platform to implement Text-To-Speech conversion.

Keywords- TTS, intel atom processor

I.INTRODUCTION

The artificial production of human speech is called Speech Synthesis. A system designed for this motive is called a Speech synthesizer, and can be implemented in both hardware and software products. A text-to-speech (TTS) system converts normal human language text into speech. A text to speech system allows people with reading disabilities or visual impairments to listen to written works on a system.

The text to speech conversion involves several steps. These can be considered as front end steps and back end steps. The implementation of all these steps is explained in this paper. The text to speech conversion steps are implemented on Intel atom processor and on laptop with i5 processor. A comparison of results on both these processors is presented in this paper.

II.DEFINITIONS

TTS: It is the abbreviation for text-to-speech. It is a software program that converts a human language text that is stored in electronic form especially in a computer system to speech.

Phoneme: It is the smallest unit of speech that is distinctive in nature and can form words of different meanings.

Phonetics: It is the part of the study of Linguistics that deals with the ways and nature of sounds of spoken human language (speech).

Prosody: It is the change in level of pitch and volume along with speed during speaking sentences. For example, the pitch generally goes down in case of a general sentence and goes up in case of a question when the end of a sentence is reached.

II.TEXT-TO-SPEECH

The TTS conversion involves several steps for conversion. The different steps used can vary from

one TTS system to another. Mostly the front end steps remain same. The back end steps vary in synthesis algorithms. The language related processing is carried on in front end steps. The speech related signal processing is carried on in back end steps.In this paper an open source Text-To-Speech system called FLITE[2] is taken as reference and its different steps of TTS are implemented on IFC946X Intel atom processor. The complete steps for implementing the Text-To-Speech system[1]can be shown as in Fig1. The steps implemented for TTS can be explained as below.

A. Text Tokenization:

The process of breaking a stream of text into words, phrases, symbols or other meaningful elements called tokens is called tokenization.Eg: Input text = "India is my country."

Token[1]=India, Token[2]=is, Token[3]=my, Token[4]=country

B. Text Normalization:

In this step, the tokens are normalized. The process of identifying numbers, abbreviations, acronyms and transforming them into corresponding words as needed, usually depending on the context of the sentence.

Eg: Token=21-05-2015

Normalized Token=twenty first may two thousand fifteen

Token=Dr. Kamal

Normalized Token=Doctor Kamal

C. Parts of Speech Tagging:

For each token, the parts of speech is identified and marked in this step. The process of assigning parts-of-speech to a word in a sentence is called parts of speech tagging[3]. The pronunciation of any word depends on the type of parts of speech it being used in a particular sentence. This step is important to get the correct pronunciation of the words by identifying the parts of speech[4].

Eg: This good(n) is very good(adj) .

In this sentence the noun word good is pronounced differently than the adjective word good.

D. Phrasing:

The entire text is divided into phrases and the phrase marking is performed at the end of each phrase in this step. The further steps like pause insertion and duration modeling will be implemented based on this step.

Eg: If youstudy well, you will getgood marks.

Phrase[1]=If you study well

Phrase[2]= you will get good marks

E. Lexical Insertion:

The words are converted into corresponding phones in this step. The word to phone conversion is carried on based on the information of parts of speech tagging.

Eg: Token=welcome

Phones={w, eh, l, k, ah, m}

The word to phone conversion is performed by applying letter to sound rules and database of phones for commonly used words. A word is searched in the database and if found its corresponding phones are returned. Else letter to sound rules are applied on the word to get the phone information[5]. The letters to sound rules are formed using various methods.

F. Pause Insertion:

In this step, a pause is inserted at the end of each phrase mark. The phrase marking information done in the phrasing step is used in this step. A pause symbol {pau} is appended at the end of each phrase mark.

Eg: { If you study well , you will get good marks } phrase will be

{ If you study well } {pau} {you will get good marks }

Pause insertion helps in the utterance of text into meaningful sentences.

G. Prosodic Tagging:

In this step, the stress and pitch variations of a word are marked.

H. Duration Modeling:

In this step the duration of the sound for each phone is marked. This step is useful for concatenative synthesis.

I. F0 Modeling:

In this step the fundamental frequency information is added to each phone. The synthesis algorithm used for TTS implementation is clustergen synthesis algorithm . In this algorithm duration modeling and F0 modeling parameters are clustered into a database and are used while synthesis operation is being performed.

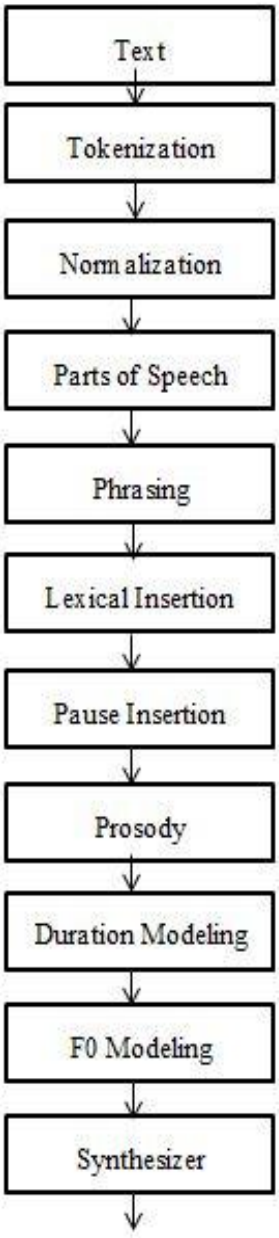

Fig 1: The block diagram showing steps in text to speech conversion

J. Synthesizer:

In this step sound will be generated based on the parameters that are marked for the given text. The synthesis algorithm used in our implementation is clustergen algorithm.

III.IMPLEMENTATION

The intel atom processor IFC946X is used for the implementation of the text to speech conversion system. The intel atom processor has been chosen as a platform for the project because of its low cost, low power and high speed operations.

The intel atom processor supports 512MB at DDR2 800 MHz memory. It supports on chip GPU upto 400 MHz.

The binary file that implements the entire text to speech system should be compatible with this processor. The generation of the binary file is done with the compilation of source code with the intel atom processor.

The source coding of different steps of the text to speech system is done using the source code of open source FLITE text to speech system. Changes to the source code of FLITE are done so that the microcontroller executes the program without any errors. Significant modifications were done to change the default utterance speed through stretch, change in the output code etc. Without these modifications, the execution was not successful and was not perceptible for the readers.

IV.RESULTS

The results of the implementation of TTS on intel atom processor are observed. Also a comparison is performed on the results of implantation of TTS on intel atom processor with a laptop with i5 processor.

The text to speech conversion output is tested with many input sentences. In most cases, a commonly spelt sentence is well understood by the listeners.

K.Input text "Hello":

The word "hello" is given as input for the intel processor and i5 processor and the results of the output are as shown in Fig 2 and Fig 3.

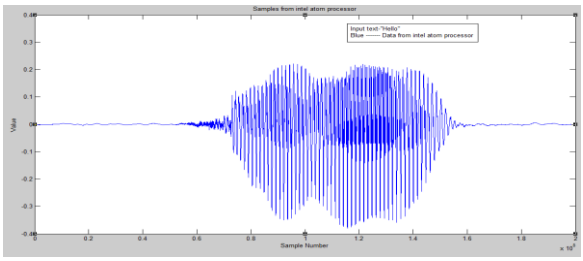

Fig 2:Outputs from intel atom processor with stretch =1.0

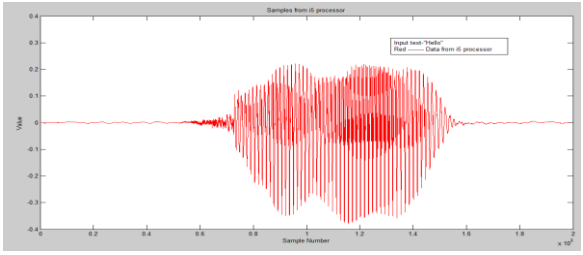

Fig 3: Outputs from i5 processor with stretch =1.0

L.Input text "G V P college of Engineering":

Now a simple sentence is given as input to both the systems. The outputs of the system are as shown in Fig.4.

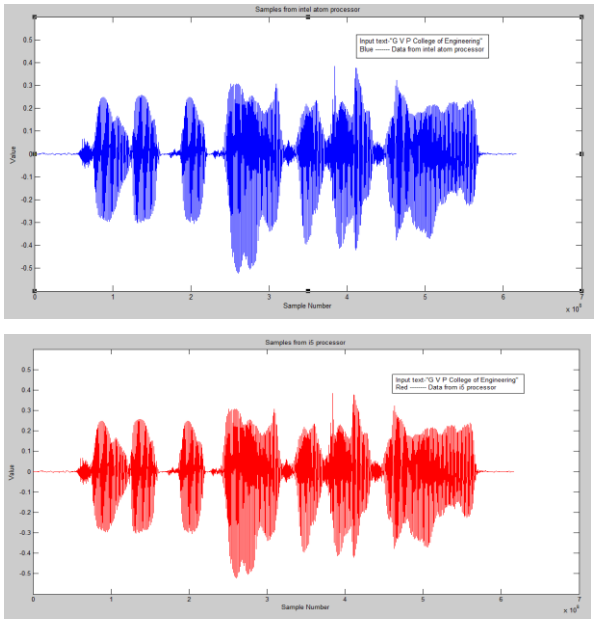

Fig 4: Outputs from intel atom processor and i5 processor with stretch =1.0

N. Input text "G V P college of Engineering"with stretch and without stretch:

Now a simple sentence is given as input to the system. A comparison of output with different stretches is shown in Fig.5.

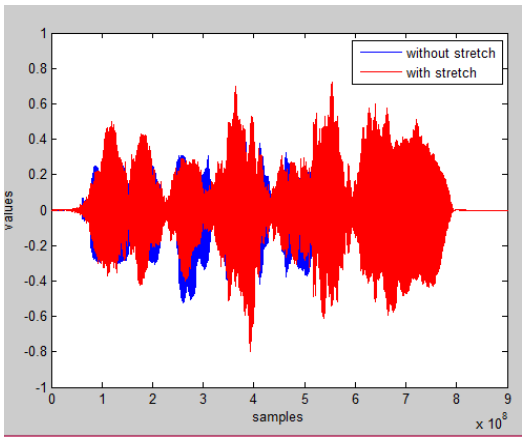

Fig 5: Outputs from intel atom processor with stretch and without stretch

V.CONCLUSION

This paper described a process of implementing the text to speech conversion system on an Intel atom processor. The results of the implementation of both intel atom processor and laptop with i5 processor are presented. From these results, it is observed that the i5 processor implementation and intel atom processor based implementation are almost similar.

The open source linux operating system is used in the implementation. Hence alow cost text to speech conversion embedded system can be built which can be used as aid for visually impaired people as an assistive tool for reading the digital text.

The text to speech conversion can be extended for accessing the email accounts for the visually impaired. This can be implemented as a web based application.

Also the text to speech conversion can be extended for printed material data instead of digital text input.

REFERENCES

[1]. Abdul Rawoof, Kulesh, Kailash Chandra Ray, "*ARM based implementation of Text-to-Speech for real time Embedded System*", in Proceedings of IEEE International Conference on Signal and Image Processing,pp. 192–196,2014.

[2]. The FLITE website. [Online]. Available : http://www.speech.cs.cmu.edu/flite/

[3]. Taylor, P. and Black, A. " *Assigning Phrase Breaks from part-of-speech Sequences"* Computer Speech and Language 12, 99-117,1998.

[4]. DeRose, S. "*Grammatical category disambiguation by statistical optimization*" Computational Linguistics, pp. 31-39, 1988.

[5]. Black, A., Lenzo, K. and Pagel, V. "*Issues in Building General Letterto Sound Rules"* 3rd ESCA Workshop on Speech Synthesis, pp. 77-80,1998.