An efficient hardware architecture for HMM-based TTS system

- **Su Hong Kiet**
- **Huynh Huu Thuan**
- **Bui Trong Tu**

University of Sciences, VNU-HCM

(Received on December 05 th 2014, accepted on September 23rd 2015)

ABSTRACT

This work proposes a hardware architecture for HMM-based text-to-speech synthesis system (HTS). In high speed platforms, HTS with software core-engine can satisfy the requirement of real-time processing. However, in low speed platforms, software core-engine consumes long time-cost to complete the synthesis process. A co-processor was designed and integrated into HTS to accelerate the performance of system.

***Keywords**: text-to-speech synthesis, HMM, HTS, SoPC, FPGA.*

INTRODUCTION

A HTS consists two parts of training part and synthesis part as shown in Fig. 1. In the training part, a context-dependent HMM database is trained from a speech database. The trained context-dependent HMM database consists of models for spectrum, pitch and state duration; and decision trees for spectrum, pitch and state duration. Then, the trained context-dependent HMM database is used by the synthesis part to generate the speech waveform from the given text.

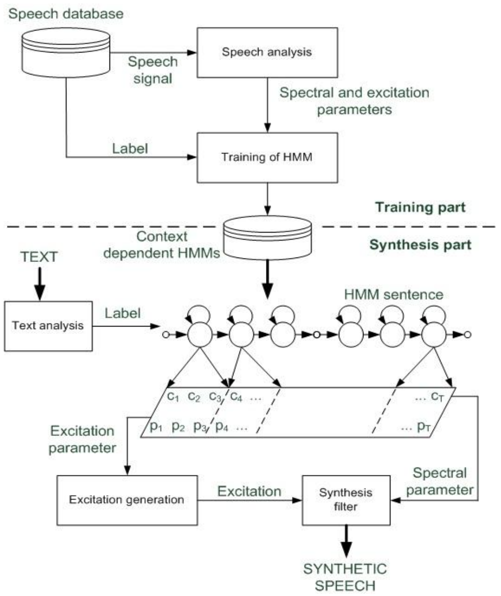


Fig. 1. Scheme of HTS

In the synthesis part, the given text is analyzed and converted into label a sequence. According to the label sequence, an HMM sentence is constructed by concatenating HMMs taken form the trained HMM database. And then, excitation and spectral parameters are extracted from HMM sentence. The extracted excitation and spectral parameters are fed to a synthesis filter to synthesize speech waveform. Depending on the fact that the spectral parameter is presented as mel-cesptral coefficients or mel-generalized cepstral coefficients, the synthesis filter is constructed as an MLSA filter or an MGLSA filter, respectively.

In recent research, HTS is applied to many languages such as Japanese [1], English [1], Korean [13], Arabic [14] and so on. Moreover, thank to the small-size of the core-engine, HTS can be implemented on various devices such as personal computer, server and so on. On high speed platforms such as PC, HTS with software core-engine can satisfy the requirement of the real-time processing. In contrast, on low speed platforms, software core-engine consumes long time-cost to convert text to speech, i.e., the system does not meet real-time processing. In order to implement an efficient HTS on low speed platforms, speeding up the performance of the core-engine is on demand. This work uses a co-processor to accelerate the performance of HTS built on FPGA-based platform.

Furthermore, the resource in low-cost system is usually limited. So the training part of the HTS is removed to reduce the bulkiness of the system. As presented above, the training part and the synthesis part are separated. Instead of integrating the training part, an offline trained HMM database is used.

The rest of this paper is organized as follow: Section 2 presents the co-processor for HTS, section 3 proposes a hardware architecture for HTS built on FPGA-based platform. Section 4 presents the experiment for evaluating the performance of the proposed system.

CO-PROCESSOR FOR HTS

HTS Working Group has been developing a software core-engine for HTS (HTS-engine) [10]. The HTS-engine provides functions to generate speech waveform from label sequence by using a trained context-dependent HMM database. The process of the generating speech waveform from label sequence can be split into three steps as follow:

•**Step 1:** parsing label sequence and creating the HMM sentence.

•**Step 2:** generating speech parameters from HMM sentence.

•**Step 3:** generating speech waveform (synthesized speech) from speech parameters.

The evaluation for the performance of the HTS-engine on various platforms shows that the time-cost for Step-1 is small, while Step-2 and Step-3 consume about 10% and 90% of the total time-cost, respectively [15]. The performance of the HTS-engine on FPGA-based platform is shown in Table 1.

Table 1. Performance of the HTS-engine on FPGA-based platform

System configuration	FPGA device	Altera CycloneIV 4CE115 FPGA chip
	CPU	Nios-II with -Floating point hardware -Instruction cache: 4KB -Data cache: 2KB
	Frequency	125 MHz
	Instruction storage	SRDAM
	Data storage	SDRAM
		Flash memory for storing trained HMM database
Synthesized speech	144,240 samples which correspond to 3.005s of speech. (Note: sampling rate is set as 48 KHz)	
Time-cost (s)	Step 1	0.25
	Step 2	2.77
	Step 3	34.27

Table 1 shows that the time-cost in FPGA-based platform is much larger than the length of the synthesized speech (above ten times). In order to accelerate the system performance, a co-processor is designed to take place the HTS-engine to carry out Step-2 and Step-3. Step-1 is still carried out by the HTS-engine to maintain the flexibility of the system. The architecture of the co-processor is shown in Fig. 2.

The speech parameter generator (SPG) carries out the processing of generating speech parameters from means and variances of states in the constructed HMM sentence. The detailed architecture of the SPG is shown in Fig. 3 A. The SPG consists of an arbiter and five sub-modules. The arbiter communicates with the main CPU via Avalon bus and controls the operation of the sub-modules via an internal bus. Each sub-module carries out its own specified task and is activated by the arbiter. After a sub-module completes its task, it informs the arbiter. And then, the arbiter deactivates the sub-module.

The synthesized speech generator (SSG) carries out the processing of generating synthesized speech from speech parameters. Similar to the SPG, the SSG consists of an arbiter and several sub-modules. The arbiter communicates with the main CPU via Avalon bus and controls the operation of the sub-modules via an internal bus. Each sub-module carries out its own specified task and is activated by the arbiter. After a sub-module completes its task, it informs the arbiter. And then, the arbiter deactivates the sub-module. The detailed architecture of the SSG is shown in Fig. 3B.

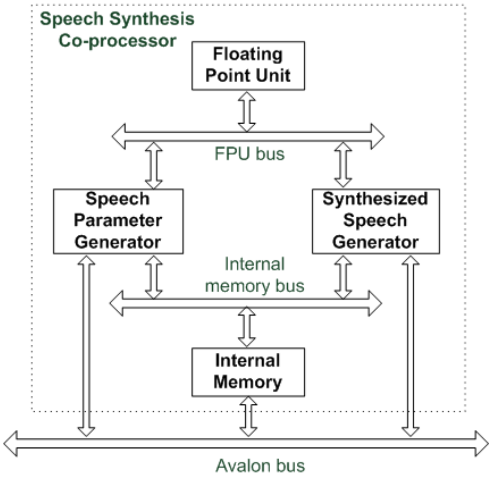
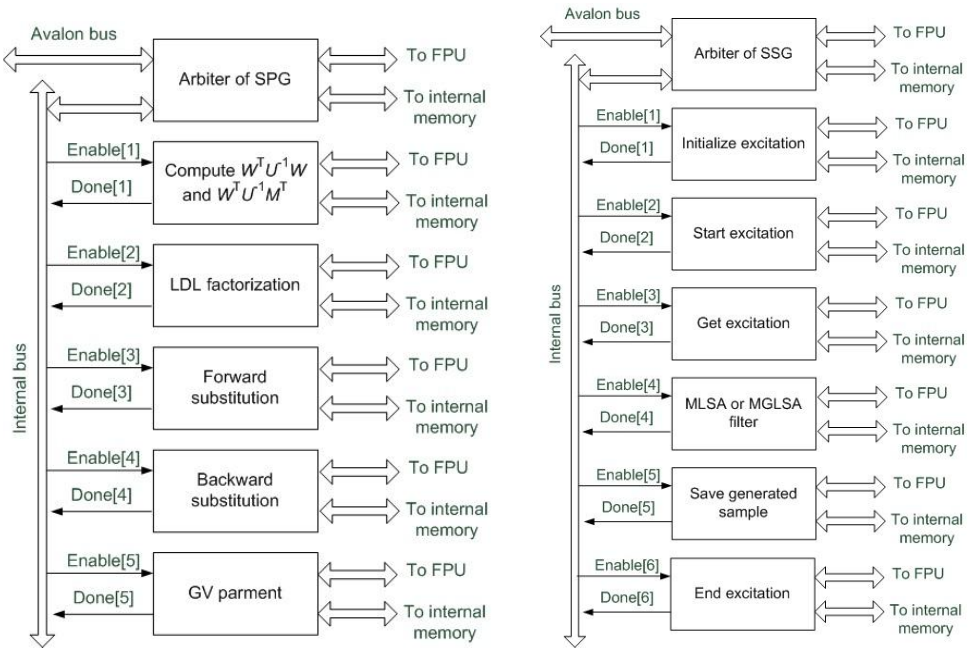

Speech Synthesis Co-processor
Floating Point Unit
FPU bus
Speech Parameter Generator
Synthesized Speech Generator
Internal memory bus
Internal Memory
Avalon bus

Fig. 2. Architecture of co-processor



(A) (B)

Fig. 3. Architecture of SPG (A) and SSG (B)

The floating point unit (FPU) is integrated into the co-processor to support the SPG and SSG to carry out operations in floating point numbers. The FPU supports operations of addition, subtraction, multiplication, division, modulo, comparison, exponential, natural logarithm and cosine. The FPU is shared for the arbiters and sub-modules of the SPG and SSG. In order to avoid the conflict, at any time, at most one arbiter or one sub-module can use the FPU, i.e., other arbiters and sub-modules must release the FPU interface bus.

The internal memory stores data which are used or created by the SPG and SSG. Similar to the FPU, the internal memory is a shared resource. At any time, at most one arbiter or one sub-module can access the internal memory, i.e., other arbiters and sub-modules must release the internal memory interface bus.

HARDWARE ARC HITECTURE FOR HTS

Fig. 4 shows the hardware architecture for HTS built on FPGA-based platform, in which a co-processor is integrated into the system to accelerate the system peformance. The Nios-II CPU is the main CPU of the system. The SDRAM is the instruction storage and data storage of the system. The PLLs are used for setting the clock frequency of the system. The UART port is used for debug mode. This architecture consists of the synthesis part of HTS only, i.e., it does not consist of the training part. So the proposed system need a trained context-dependent HMM database. Since the HMM database is saved in files, a flash memory is used to store the HMM database so that we can use the read only zip file system (which is supported by Altera) to load data from the HMM database.

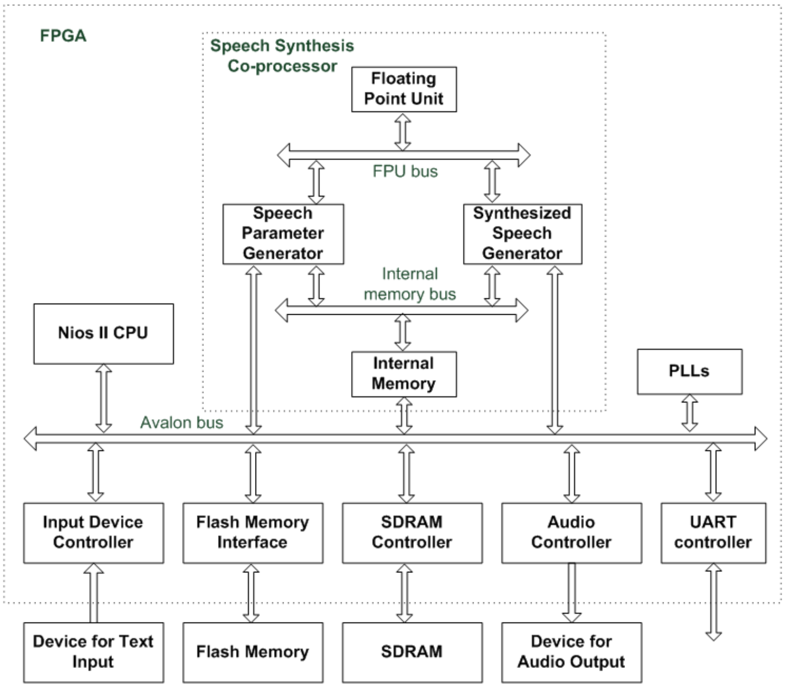


Fig. 4. Hardware architecture for HTS

EXPERIMENT

The proposed system is shown in Fig. 4 on Stratix IV FPGA development board, in which the input text device is a touch-screen and the audio output device is a DAC card connecting to a speaker. The performance of the system is shown in Table 2.

Table 2 shows that the performance time-cost is smaller than the length of the synthesized speech, i.e., the requirement of real-time processing is met. Comparing to the system which does not have the co-processor, the performance time-cost is reduced significantly. When co-processor is not used, the performance time-cost is above ten times larger than the length of synthesized speech. But after integrating co-processor into the system and setting the system configuration appropriately, the performance time-cost can be reduced to a value smaller than the length of the synthesized speech.

Table 2. Performance of the HTS on FPGA-based platform with a co-processor

Input text	Synthesized speech (Sampling rate = 38 KHz)		Time-cost (s)
	Number of samples	Length (s)	
Bộ Giáo dục và Đào tạo	95040	2.501	2.462
Đại học khoa học tự nhiên	95040	2.501	2.428
Đại học tự nhiên	74880	1.970	1.882
Thuê bao vừa được gọi không liên lạc được	116640	3.069	3.040
Thành phố Hồ Chí Minh ngày mùng hai tháng chín	128460	3.381	3.375

Moreover, the synthesized speech is intelligible and has the same quality to the speech which is synthesized by HTS built on PC-platform. Denoting waveforms which generated from the same input text by the proposed HTS and the HTS built on PC-platform by X_1 and X_2, respectively.

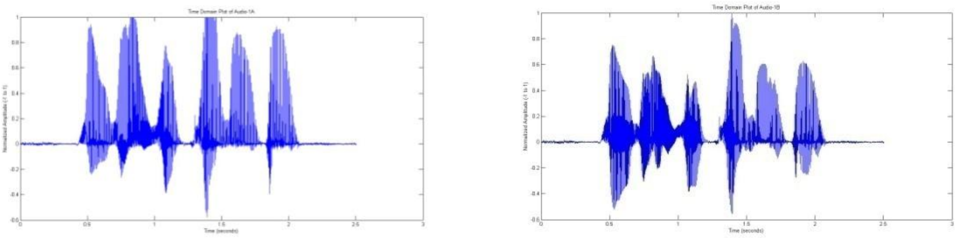
$$X_1 = [x_{11}, x_{12}, \dots, x_{1N}]$$

$$X_2 = [x_{21}, x_{22}, \dots, x_{2N}]$$

where x_{1i} and x_{2i} with $i = 1,2,\dots,N$ are samples of X_1 and X_2, respectively.

The mean square error (MSE) between two vectors X_1 and X_2 is calculated as the following equation

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_{1i} - x_{2i})^2 \qquad (1)$$



A

B

Fig. 5. Waveform generated from the input text ”bộ giáo dục và đào tạo” by proposed HTS (A) and HTS built on PC-platform (B)

Applying Eq.-1 to waveforms which are generated from different input text, we obtain the result in Table 3.

Table 3. Mean square error between waveforms generated by proposed HTS and HTS built on PC-platform

Input text	MSE
Bộ Giáo dục và đào tạo	0.034
Đại học khoa học tự nhiên	0.020
Đại học tự nhiên	0.022
Thuê bao vừa được gọi không liên lạc được	0.045
Thành phố Hồ Chí Minh ngày mùng hai tháng chín	0.038

Table 3 shows that the MSEs between waveforms generated by two systems are smaller than 4.5 %, i.e., waveforms generated from the two systems are alike.

CONCLUSION

An efficient hardware architecture for HTS built on FPGA-based platform was proposed by this work. In the proposed architecture, a co-processor is used to accelerate the performance of the system. The experiment results show that using a co-processor can reduce the performance time-cost significantly. It leads the system meeting the requirement of real-time processing. Moreover, the speech synthesized by the proposed system is intelligible and has a waveform alike to the one which is generated by the HTS built on PC-platform.

Một kiến trúc phần cứng hiệu quả cho hệ thống TTS trên cơ sở HMM

- **Sú Hồng Kiệt**
- **Huỳnh Hữu Thuận**
- **Bùi Trọng Tú**

  Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM

TÓM TẮT

Bài báo này đề xuất một kiến trúc phần cứng cho hệ thống tổng hợp tiếng nói từ văn bản trên cơ sở HMM (HTS). Trên những nền tảng có tốc độ cao, hệ thống HTS với engine tổng hợp được xây dựng bằng phần mềm có thể thỏa mãn yêu cầu về xử lý thời gian thực. Tuy nhiên, trên những nền tảng có tốc độ thấp, engine bằng phần mềm tốn nhiều thời gian để hoàn tất quá trình tổng hợp. Do đó, một bộ đồng xử lý (co-processor) đã được thiết kế và tích hợp vào hệ thống HTS nhằm gia tăng hiệu năng của hệ thống.

Từ khóa*: text-to-speech synthesis, HMM, HTS, SoPC, FPGA.*

REFERENCES

[1]. K. Tokuda, H. Zen, A.W. Black, An HMM-based speech synthesis system applied to English, *In Speech Synthesis, Proceedings of 2002 IEEE Workshop on, IEEE*, 227-230 (2002).

[2]. K. Tokuda, T. Masuko, N. Miyazaki, T. Kobayashi, Multi-space probability distribution HMM, *IEICE TRANSACTIONS on Information and Systems*, 85, 3, 455-464 (2002).

[3]. K. Tokuda, T. Masuko, N. Miyazaki, T. Kobayashi, Hidden Markov models based on multi-space probability distribution for pitch pattern modeling, *In Acoustics, Speech, and Signal Processing, Proceedings., 1999 IEEE International Conference*, 1, 229-232 (1999).

[4]. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, T. Kitamura, Duration modeling for HMM-based speech synthesis, *In ICSLP*, 98, 29-31 (1998).

[5]. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, T. Kitamura, Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. *In Sixth European Conference on Speech Communication and Technology* (1999).

[6]. K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, T. Kitamura, Speech parameter generation algorithms for HMM-based speech synthesis, In Acoustics, *Speech, and Signal Processing, ICASSP'00. Proceedings. 2000 IEEE International Conference*, 3, 1315-1318 (2000).

[7]. T. Fukada, K. Tokuda, T. Kobayashi, S. Imai, An adaptive algorithm for mel-cepstral analysis of speech, In Acoustics, speech, and signal processing, 1992. ICASSP-92., 1992 IEEE International Conference on, 1, 137-140 (1992).

[8]. K. Tokuda, T. Kobayashi, T. Masuko, S. Imai, Mel-generalized cepstral analysis-a unified approach to speech spectral estimation, *In ICSLP* (1994).

[9]. SPTK Working Group. (2013, December). Reference manual for speech signal

processing toolkit Ver 3.7. http://sp-tk.sourceforge.net/

[10]. HTS Working Group. HMM-based speech synthesis engine (hts_engine API) Ver. 1.06. http://htsengine.sourceforge.net/

[11]. N.M. Pham, D.N. Dau, Q.H. Vu, Distributed web service architecture towards robotic speech communication: A Vietnamese case study, *Int. J. Adv. Robotic Sy*, 10, 130 (2013).

[12]. P. Taylor, Text-to-speech synthesis, *Cambridge University Press* (2009).

[13]. S.J. Kim, J.J. Kim, M. Hahn, HMM-based Korean speech synthesis system for hand-held devices. Consumer Electronics, *IEEE Transactions on*, 52, 4, 1384-1390 (2006).

[14]. K.M. Khalil, C. Adnan, Arabic HMM-based speech synthesis. In Electrical Engineering and Software Applications (ICEESA), *2013 International Conference*, 1-5 (2013).

[15]. H.B. Nguyen, T.B.T. Cao, T.T. Bui, H.T. Huynh, A performance evaluation of HMM based text- to- speech system on various platforms, *Proceedings of ICDV-2013*, 265-267 (2013).