ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

MAXIMUM SPEED PROCESSING OF SPEAKER VERIFICATION SYSTEM IMPLEMENTED USING VFPU

R.Rajathilagam[1] ,K.Priya[2]

M.E VLSI Design,Department of Electronics and Communication Engineering [1]
Assistant Professor,Department of Electronics and Communication Engineering [2]
TRP Engineering College(SRM Group),Trichy,Tamil Nadu,India.
rajathilagam93@gamil.com

***Abstract*— The project idea is to implement the speaker verification system on Field programmable gate array (FPGA). The design of the VFPU should be performed in order to ensure the capacity of the system to work in real time and to speedup the resolution of any vector floating-point operation involved in verification algorithm. The verification process was carried out by configuring Microblaze with the scalar floating-point unit provided by Xilinx. The experimental result show that when comparing our proposed system against both systems, the number of clock cycles is reduced. The main advantage of the VFPU is its flexibility, which allows quick adaptation of the software to the potential changes produced in both the system and the user requirements. The algorithm was tested over a public database that contains the utterances of different users acquired under different environmental conditions, providing good recognition rates. Throughput is increased by reducing the execution time.**

KEY WORDS: **Xilinx,Vector Floating-Point Unit (VFPU), Hardware-Software codesign, Speaker verification.**

I.INTRODUCTION

SPEAKER verification system over the telephone has attracted much attention, primarily because of the rapid increase in the electronic banking and electronic commerce. Although substantial progress in telephone-based speaker verification has been made, two issues have making difficulties on something to happen the pace of development. First, the quality or condition to mobile phone variations remains a challenge: transducer variability could result in relating to sound or sense of hearing mismatches between the speech data gathered from different mobile phone. Second, the accuracy of mobile phone identification is a concern: a wrong identification for the mobile phone used by the speaker can result in wrong mobile phone compensation. To enhance the quality of these speaker verification systems, handset compensation and identification techniques are able to replacable.

The verification process is achieved by making the sound or pronouncing statements in several seconds in length. The flexibility of a software implementation happening in a short time development of applications using fixed hardware architectures. Note that any change introduced in any of the parameters that leads to desiging the entire system again or change the techniques at different ways. These limitations can be partially achieved by adding a **floating-point unit** (FPU) as part of the FPGA design. However, most of the proposed FPUs only include basic arithmetic operations (mul, add, sub and div).

Although the FPU is able to find the solution for these computations, the time needed for their calculation could be sufficiently accelerated if a **vector floating-point unit** (VFPU) is used. The system consists of the Xilinx, and a VFPU that calculates any vector operation defined in floating-point format. The architecture of the VFPU is relating to a generic, so that it can be easily suitable for other soft-core microprocessors or FPGA families. The main feature of the VFPU is its flexibility, which provides the possibility of easily introducing modifications in the algorithm or including new processing stages. In the some case of a speaker verification system, such flexibility allows designing the VFPU easily and number of bits used for the results of arranging in a systematic form or codes of the input samples or the number of coefficients included in the feature vector. Furthermore, in applications in which samples of voice are affected by environmental conditions (background noise, distortion, etc.), or users have a remarkable common characteristic in their voice (due to age, prosodic features, etc.), changes in the parameters can be quickly introduced for adapting the system to such particular characteristics in order to improve the recognition rates.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

II. VFPU ARCHITECTURE

VFPU stands for Vector Floating Point Unit. As its name indicates it's a FPU with the ability to compute vector operations such as cross product, vector-matrix transform, matrix multiply and arithmetic operations.

A.FPU

A F**loating-Point Unit** (**FPU**) is a part of a computer system specially designed to carry out operations on floating point numbers. Typical operations are addition, subtraction, multiplication, division, square root, and bitshifting. Some systems can also perform various transcendental functions such as exponential or trigonometric calculations, though in most modern processors these are done with software library routines.

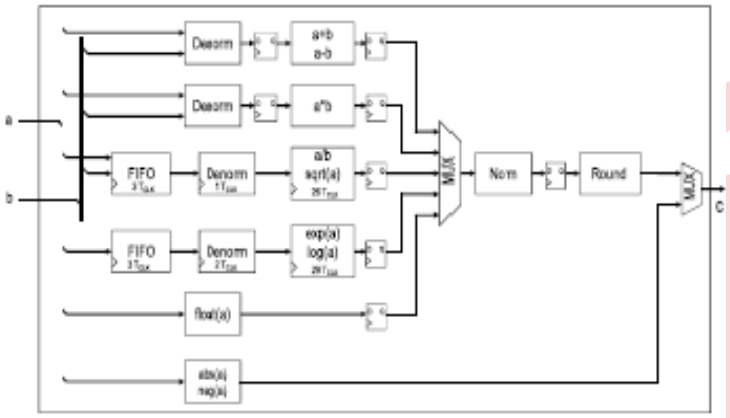

Fig.1 Internal Architecture of FPU

FPU stands for **Floating Point Uni**t. The FPU is designed in order to perform the operations. Its internal design includes a specific block capable of performing the exponential function, which is the basis of the kernel employed by the SVM classifier. The addition of this block, as part of the VFPU, is necessary to solve the algorithm in a real time.

Implementing floating-point arithmetic in hardware can solve two separate problems. First, it greatly speeds up floating-point arithmetic and calculations. Implementing a floating-point instruction will require at a generous estimate at least twenty integer instructions, many of the conditional operations, and even if the instructions are executed on an architecture which goes to great lengths to speed up execution, this will be slow. In contrast, even the simplest implementation of basic floating-point arithmetic in hardware will require perhaps ten clock cycles per instruction, a small fraction of the time a software implementation would require.

FPU performs only the basic arithmetic operation of addition,subtraction,multiplication,division,exponential,normalization and rounding operations. If any changes in the parameters of system we have to redesigne the whole system or techniques.

Although the FPU is able to resolve these computations, the time needed for their calculation could be significantly accelerated if a vector floating-point unit (VFPU) is used. When operating with vectors, the VFPU increases the throughput by reducing both the number of CPU fetches and the number of memory accesses.

B.VFPU Description

The internal architecture of VFPU is designed to make the best vector computations based on the execution of a set of basic floating-point operations. In this way, these computations can be performed without unnecessary access to the external memory, which are used to store temporary results.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

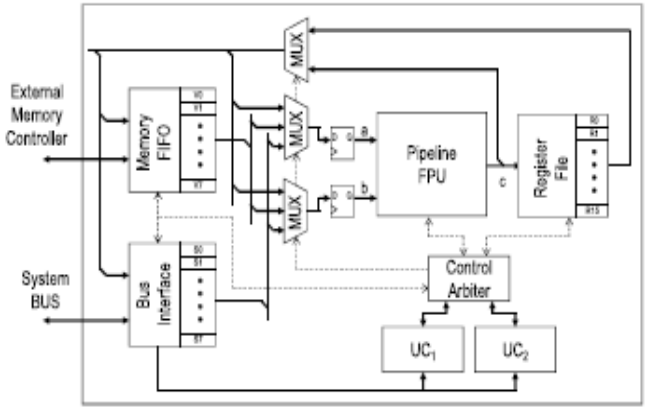

Fig.2 Data path description for the VFPU

VFPU consist of register file which is used to read or write the scalar or vector operands and memory addresses and the vectors are located at the register file.The register file stores the result provided by the FPU. Memory FIFO reads vectors stored from the external memory. Fig 1. Shows the internal architecture of FPU mentioned in data path description as shown in fig 2.

FPU performs both the arithmetic and exponential operations, executed by using number of clock cycles. VFPU consist of control arbiter and two control units. In earlier days VFPU consist of only one control unit. In this paper we use two control units **UC1**(control unit 1) controls the calculation of the exponential operations. **UC2**(control unit 2) manages the evaluations of the exponential operation. Both control units try to access the FPU. Control arbiter is needed to manage the permission. The design of VFPU should be performed in real time work. The VFPU design includes two control units that maximize the throughput and reduce the number of clock cycles.

III. SIMULATION RESULT

Using only one control unit the exponential function (including the accumulation) are solved in 312·*T*CLK and 41· CLK, respectively as shown in fig.3. However, if both operations are launched in parallel, the execution time is mainly dominated by the calculation of exponent, which is a longer operation.

Using two control unit the exponential function (including accumulation) are solved in 312·*T*CLK, respectively as shown in fig.4.

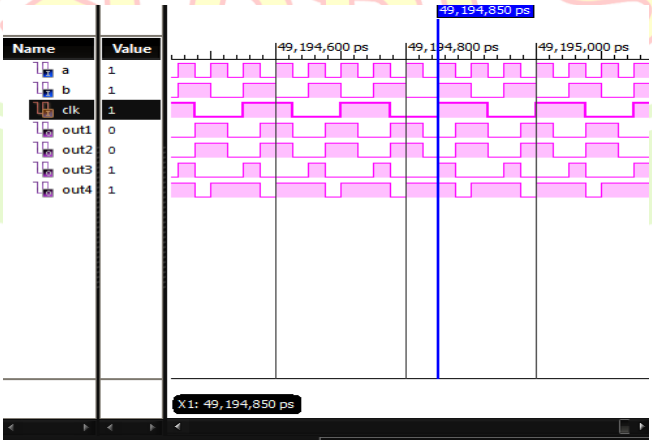

Fig.3 VFPU Operation by using Only One Control Unit

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Special Issue 10, March 2016

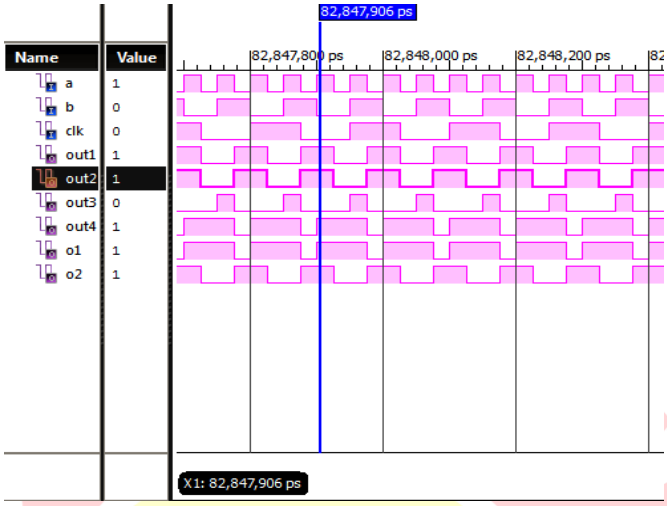

Fig.4 VFPU Operation by using Two Control Unit

IV.EXPERIMENTAL RESULTS

In order to prove our aim Xilinx has been selected for implementation. The clock cycle reduction is done by using two control units of VFPU. The control units of VFPU is designed from the verilog behavioral modeling of Hardware Description Language. The UC1 consist of basic arithmetic operations of addition, subtraction, multiplication operation executed by using 4 clock cycles and exponential operation executed by using 37 clock cycles. UC2 consist of exponential operation and addition operation.

V. CONCLUSION

This paper gives the design and implementation on xilinx. It also describes a generic architecture of VFPU that solves the vector floating-point computations. Additionally, the architecture provides a high flexibility, which allows the quick adaptating the parameters. It design includes two control units that reduce the number of clock cycles.

REFERENCES

[1] J. P. Campbell, Jr., "Speaker recognition: A tutorial," Proc. IEEE,vol. 85, no. 9, pp. 1437–1462, Sep. 1997.
[2] D. A. Reynolds, "An overview of automatic speaker recognition technology," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), vol. 4. May 2002, pp. 4072–4075.
[3] L. Jia and B. Xu, "Including detailed information feature in MFCC for large vocabulary continuous speech recognition," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), vol. 1. May 2002, pp. I805–I808.
[4] D. G. Childers, D. P. Skinner, and R.C. Kemerait, "The cepstrum: A guide to processing," Proc. IEEE, vol. 65, no. 10, pp. 1428–1443, Oct. 1977.
[5] D.-P.Munteanu and S.-A. Toma, "Automatic speaker verification experiments using HMM," in Proc. 8th Int. Conf. Commun. (COMM), Jun. 2010, pp. 107–110.
[6] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," IEEE Trans. Speech Audio Process., vol. 3, no. 1, pp. 72–83, Jan. 1995.
[7] A. Ganapathiraju, J. E. Hamker, and J. Picone, "Applications of support vector machines to speech recognition," IEEE Trans. Signal Process., vol. 52, no. 8, pp. 2348–2355, Aug. 2004.
[8] V. Wan and S. Renals, "Speaker verification using sequence discriminant support vector machines," IEEE Trans. Speech Audio Process., vol. 13, no. 2, pp. 203–210, Mar. 2005.
[9] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," Data Mining Knowl. Discovery, vol. 2, no. 2,pp. 121–167, 1998.
[10] V. Wan and W. M. Campbell, "Support vector machines for speaker verification and identification," in Proc. IEEE Signal Process. Soc. Workshop Neural Netw. Signal Process. X, vol. 2. Dec. 2000, pp. 775–784.
[11] M. Fons, F. Fons, and E. Cantó, "Fingerprint image processing acceleration through run-time reconfigurable hardware," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 57, no. 12, pp. 991–995, Dec. 2010.
[12] M. López, J. Daugman, and E. Cantó, "Hardware–software co-design of an iris recognition algorithm," IET Inf. Secur., vol. 5, no. 1, pp. 60–68, Mar. 2011.
[13] W.-Y. Choi, D. Ahn, S. B. Pan, K. I.Chung, Y.Chung, and S.-H. Chung, "SVM-based speaker verification system for matchon-card and its hardware implementation," ETRI J., vol. 28, no. 3,pp. 320–328, Jun. 2006.
[14] J.Manikandan, B. Venkataramani, and V.Avanthi, "FPGA implementation of support vector machine based isolated digit recognition system,"in Proc. 22nd Int. Conf. VLSI Design, Jan. 2009, pp. 347–352.
[15] N.-V. Vu, J.Whittington, H.Ye, and J. Devlin, "Implementation of the MFCC front-end for low-cost speech recognition systems," in

Proc.IEEE Int. Symp. Circuits Syst. (ISCAS), May/Jun. 2010, pp. 2334–2337.
[16] P. EhKan, T. Allen, and S. F. Quigley, "FPGA implementation for GMM-based speaker identification," Int. J.Reconfigurable Comput.,vol. 2011, pp. 1–8, Nov. 2011.
[17] R. Ramos-Lara, M. López-García, E. Cantó-Navarro, and L. Puente-Rodriguez,"Real-time speaker verification system implemented on reconfigurable hardware," J. Signal Process. Syst.,vol. 71, no. 2, pp. 89–103, May 2013.
[18] P.Karlström, A.Ehliar, and D.Liu, "High-performance, low-latency field-programmable gate array-based floating-point adder and multiplier units in a Virtex 4," IET Comput. Digit. Techn., vol. 2, no. 4,pp. 305–313, Jul. 2008.
[19] Y. J. Chong and S. Parameswaran, "Configurable multimode embedded floating-point units for FPGAs," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 11, pp. 2033–2044, Nov. 2011.
[20] S. Chen, R. Venkatesan, and P. Gillard, "Implementation of vector floating-point processing unit on FPGAs for high performance computing," in Proc. Can. Conf. Elect. Comput. Eng. (CCECE), May 2008, pp. 881–886.
[21] J. Kathiara and M. Leeser, "An autonomous vector/scalar floating point coprocessor for FPGAs," in Proc. IEEE 19th Annu. Int. Symp. Field- Program. Custom Comput. Mach. (FCCM), May 2011, pp. 33–36.