International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 3, No. 2, May, 2014

Research Paper

RECONFIGURABLE ACCELERATOR FOR BIOMETRIC SEARCH ENGINE USING FPGA AND MATLAB

P Rajasekar[1]*, G Sundar[2], P Sebastin Ashok[1] and S Shyamaladevi[1]

*Corresponding Author: **P Rajasekar,** ✉ p.rajasekar28@yahoo.com*

Biometric technologies are becoming the foundation of an extensive array of highly secure identification and personal verification solutions. One of the most significant parts of an investigation is fingerprinting. Today, the internet and high memory computers allow law enforcement officials to place prints in a massive database run by the Federal Bureau of Investigation (FBI) that can be accessed from anywhere in the world. While we interface fingerprint sensor with computers, several constraints appear, such as memory limitations, time of computation, recognition robustness and power consumption. In order to get benefit of the performance of recent FPGA devices, such as amount of memory, number of logic blocks, low power consumption and faster clocks, this paper proposes new hardware implementation algorithms for fingerprinting technology.

Keywords: Biometric technologies, FPGA, FBI, Memory

INTRODUCTION

Biometrics are automated methods of recognizing a person based on a physiological or behavioral characteristic. Among the features measured are; face fingerprints, hand geometry, handwriting, iris, retinal, vein, and voice. Biometric technologies are becoming the foundation of an extensive array of highly secure identification and personal verification solutions. As the level of security breaches and transaction fraud increases, the need for highly secure identification and personal verification technologies is becoming apparent.

Biometric-based solutions are able to provide for confidential financial transactions and personal data privacy. The need for biometrics can be found in federal, state and local governments, in the military, and in commercial applications. Enterprise-wide network security infrastructures, government IDs, secure electronic banking, investing and other financial transactions, retail sales, law enforcement, and health and social services are already benefiting from these technologies. Fingerprint recognition is one of the most common techniques used for biometric identification. Also one of the most significant parts of an investigation is fingerprinting. Seen as a technology that is a logical replacement for antiquated and cumbersome personal identification numbers (PINs) and passwords, biometrics is a more secure alternative to enhance individual identification accuracy and system security.

Nowadays the research effort in fingerprint algorithms is focused on improving their

[1] M.E VLSI Design, ECE Department, Sembodai Rukmani Varatharajan Engineering College.

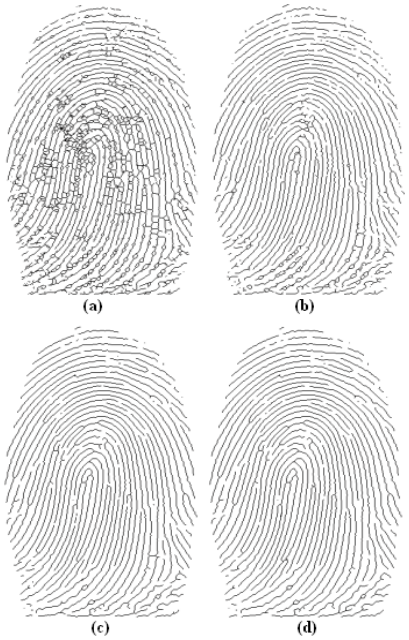
[2] Assistant Professor, ECE Dept, Sembodai Rukmani Varatharajan Engineering College Vedaranyam, Tamilnadu, India.

performances, basically increasing the reliability and reducing the error rates. Usually the implementation is based on a high-performance microprocessor, such as a desk computer, able to work at frequencies in the GHz range. The algorithm runs on a microprocessor that sequentially executes the routines involved in the fingerprint processing. Recent advances in the field microelectronics have improved the microprocessor computational power which allows algorithms to run with high accuracy without increasing the execution times. However, we still depend on computers for all external world technical interfacing devices.

FINGERPRINTS AS A BIOMETRIC

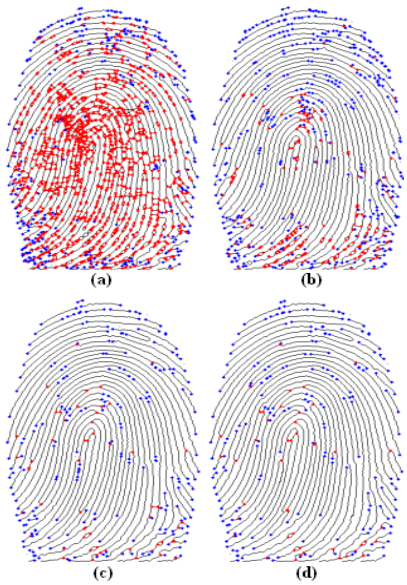
Among all biometric traits, fingerprints have one of the highest levels of reliability and have been extensively used by forensic experts in criminal investigations. A fingerprint refers to the flow of ridge patterns in the tip of the finger. The ridge flow exhibits anomalies in local regions of the fingertip (Figure 1), and it is the position and orientation of these anomalies that are used to represent and match fingerprints.

Figure 1: Fingerprint Image Thinning



Although not scientifically established, fingerprints are believed to be unique across individuals, and across fingers of the same individual. Even identical twins having similar DNA, are believed to have different fingerprints. Traditionally, fingerprint patterns have been extracted by creating an inked impression of the fingertip on paper. The electronic era has ushered in a range of compact sensors that provide digital images of these patterns. These sensors can be easily incorporated into existing computer peripherals like the mouse or the keyboard (Figure 2), thereby making this mode of identification a very attractive proposition. This has led to the increased use of automatic fingerprint-based authentication systems in both civilian and law enforcement applications.

Figure 2: Minutia Points (Bifurcation-Red, Termination-Blue)



A. Fingerprint Representation

The uniqueness of a fingerprint is determined by the topographic relief of its ridge structure and the presence of certain ridge anomalies termed as minutiae points. Typically, the global configuration defined by the ridge structure is used to determine the class of the fingerprint, while the distribution of minutiae points is used

to match and establish the similarity between two fingerprints. Automatic fingerprint identification systems, that match a query print against a large database of prints (which can consist of millions of prints), rely on the pattern of ridges in the query image to narrow their search in the database (fingerprint indexing), and on the minutiae points to determine an exact match (fingerprint matching).

B. Extraction of Minutiae and Elimination of False Minutiae

The accuracy of the extraction of minutiae from binary fingerprint image depends on the success of enhancement and thinning processes. If ridge and valley structures of the fingerprint image are damaged during these processes, it causes to extract false minutiae points. As a result, it is very important to gain best results from these processes in order to extract minutia correctly.

HAMMING DISTANCE

In information theory, the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. In another way, it measures the minimum number of substitutions required to change one string into the other, or the minimum number of errors that could have transformed one string into the other.

Algorithm Example

The Python function hamming distance() computes the Hamming distance between two strings (or other iterable objects) of equal length, by creating a sequence of Boolean values indicating mismatches and matches between corresponding positions in the two inputs, and then summing the sequence with False and True values being interpreted as zero and one.

def hamming_distance(s1, s2)

#Return the Hamming distance between equal-length sequences

if len(s1) != len(s2)

raise ValueError (“Undefined for sequences of unequal length”)

return sum(ch1 != ch2 for ch1, ch2 in zip(s1, s2))

The following C function will compute the Hamming distance of two integers (considered as binary values, that is, as sequences of bits). The running time of this procedure is proportional to the Hamming distance rather than to the number of bits in the inputs. It computes the bitwise exclusive or of the two inputs, and then finds the Hamming weight of the result (the number of nonzero bits) using an algorithm of Wegner that repeatedly finds and clears the lowest-order nonzero bit.

unsigned hamdist(unsigned x, unsigned y)

{unsigned dist = 0, val = x ^ y; // XOR

// Count the number of set bits while(val)

{++dist; val &= val - 1;}

return dist;}

PROPOSED ARCHITECTURE

Figure 3 shows the architecture of the proposed system. Grey scale eye images are captured and processed to generate fingerprint template database which will be stored on Rom. For lager databases, off-chip ROM will be interfaced to FPGA. Small databases can be stored on chip. Live template is compared with the templates in the database by calculating hamming distance between them. Since orientation of iris changes time to time best match is found by shifting the data template 2 bits left and then performs comparison operation again with the live template. If match found search will be stopped after generating the result else search will continue for the entire database. Result can be displayed on monitor or through LED an LCD. Matching module has the internal structure as follows:

- Memory Module
- HD modules
- Adder tree
- Comparator and display

Figure 3: System Architecture

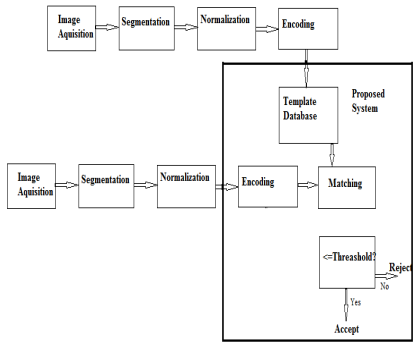
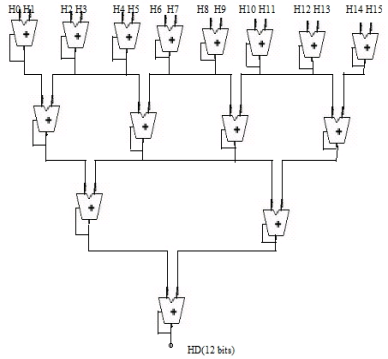


Figure 4: Adder Tree



Implementation Concepts

Sample database is created by generating the templates for 10 sample eye images taken from MMU1 database. Segmentation, Normalization and Encoding is done using matlab open source code. Matching module is implemented using verilog HDL. Proposed design is implemented using adders, shifter, and magnitude comparator. Xilinx isim (Ise 12.4) tool is used to generate simulation results. When live template is given as input, and when reset is not asserted, search starts. Search starts at first data template in the database. Live template is shifted 2 bits left and again compared with the stored template. Total 10 left shifts and 10 right shifts are performed for displacement alignment. In this way HD is calculated between live template and each database template for 21 times. If HD<=Threashold, m (match) will be asserted high.

CONCLUSION & FUTURE SCOPE

Low cost and fast execution are important parameters for real time authentication systems. The main purpose of the work described in this paper is to implement most time consuming part of fingerprint recognition algorithm on a low-cost FPGA. The proposed design is implemented using verilog HDL. This design is suitable for small low cost applications. Future improvements in the system can be done by implementing the feature extraction task on hardware which can further reduce the authentication time. Instead of using Matlab we can use Micro blaze soft core processor and can implement the iris localization and iris normalization in C. By this we can implement entire system on single chip.

REFERENCES

1. Daugman J G (2003), "The Importance of Being Random: Statistical Principles of Iris Recognition", *Pattern Recognition*, Vol. 36, pp. 279-291.
2. Daugman J G (2004), "How Iris Recognition Works IEEE Trans", *Circuits Syst. Video Technology*, Vol. 14, No. 1, pp. 21-30.
3. James Wayman, Anil K Jain, Davide Maltoni and Dario Maio (2005), "Biometric Systems: Technology, Design and Performance Evaluation", *Springer-Verlag Limited, London*.
4. Maitane Barrenechea, Jon Altuna and Miguel San Miguel (2007), "A Low Cost FPGA-Based Embedded Fingerprint Verification Embedded Fingerprint Verification and Matching System", *Fifth Workshop on Intelligent Solutions in Embedded Systems* (WISES 07).
5. P Wildes, S C Hsu, R J Kolczynski, R Matey, J C Asmuth and S E McBride (1996), "Automated, Noninvasive Iris Recognition System and Method", *U.S. Patent*, No. 5, pp. 572-596.
6. Raimond Thai (2003), "Fingerprint Image Enhancement and Minutiae Extraction", *Master's thesis, University of Western Australia*.

7. Ryan N Rakvic, Brandley J Ulis, Randy P Broussard, Robert W Ives and Neil Steiner (2009), "Parellelizing Iris Recognition, IEEE Transactions on Information Forensics and Security", Vol. 4, No. 4, December 2009.
8. Shenglin Yang, Kazuo Sakiyama, and Ingrid Verbauwhedem (2006), "Efficient and Secure Fingerprint Verification for Embedded Devices", *URASIP Journal on Applied Signal Processing*, pp. 1-11.
9. ZHOU Hu-Lin, XIE Mei (2010), "Iris Biometric Processor Enhanced Module FPGA based Design", *2010 Second International Conference on Computer Modeling and Simulation*.

International Journal of Engineering Research and Science & Technology
Hyderabad, INDIA. Ph: +91-09441351700, 09059645577
E-mail: editorijlerst@gmail.com or editor@ijerst.com
Website: www.ijerst.com