

# An ASIC Implementation of Fingerprint Thinning Algorithm

Seung-Min Jung, *Member, KIMICS*

**Abstract**—This paper proposes an effective fingerprint identification system with hardware block for thinning stage processing of a verification algorithm based on minutiae with 39% occupation of 32-bit RISC microprocessor cycle. Each step of a fingerprint algorithm is analyzed based on FPGA and ARMulator. This paper designs an effective hardware scheme for thinning stage processing using the Verilog-HDL in 160x192 pixel array. The ZS algorithm is applied for a thinning stage. The logic is also synthesized in 0.35μm 4-metal CMOS process. The layout is performed based on an auto placement-routing and post-simulation is performed in logic level. The result is compared with a conventional one.

**Index Terms**— ASIC, Verilog-HDL, thinning, fingerprint, RTL, VLSI,

## I. INTRODUCTION

A fingerprint is a collection of ridge and valley patterns of a human fingertip. Each fingerprint has its own unique characteristics, thus distinguishing each individual from others. A fingerprint verification algorithm has two phases: enrollment and verification[1]. In the off-line enrollment phase, an enrolled fingerprint image is preprocessed, and the minutiae are extracted and stored. In the on-line verification phase, the similarity between the enrolled minutiae and the input minutiae is examined. Fig. 1 shows a general fingerprint verification algorithm.

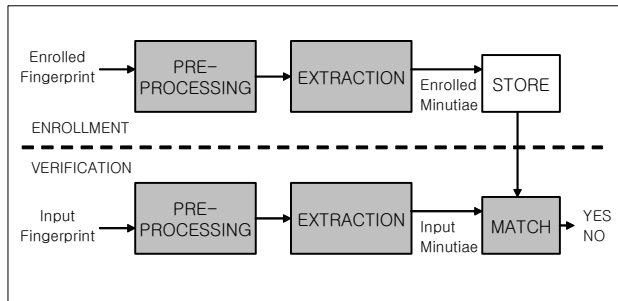


Fig. 1 Fingerprint verification concept

There are three steps involved in the verification process: 1) image preprocessing, 2) minutiae extraction, and 3) minutiae matching.

Image preprocessing refers to the refinement of the fingerprint image against the image distortion obtained from a fingerprint sensor. It consists of three stages as shown in Fig. 2.

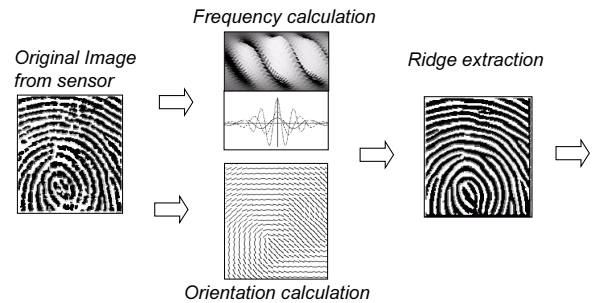


Fig. 2 Image pre-processing

The binary conversion stage applies a low-pass filter to smooth the high frequency regions of the image and threshold to each sub-segment of the image. Gabor filter is strongly recommended to get improved image and reduce the noise level[2]. It made the image more effective to extract the minutiae from it. The thinning stage generates a one-pixel wide skeleton image by considering each pixel with its neighbors. In the positioning stage, the skeleton is transformed and/or rotated so that valid minutiae information can be extracted. Minutiae extraction refers to the extraction of features in the fingerprint image. Minutiae mean ridge ending or bifurcation of fingerprint images. After this step, some minutiae are detected and stored in a pattern file, which includes the position, orientation, and type (ridge ending or bifurcation) of the minutiae. Fig. 3 shows the Thinning & Minutiae extraction

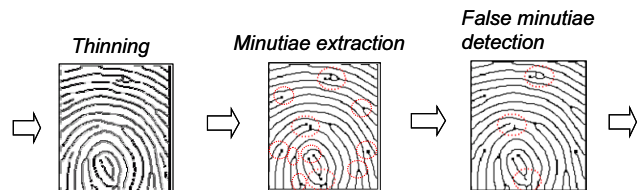


Fig. 3 Thinning & Minutiae extraction

Manuscript received November 12, 2010; revised November 22, 2010; accepted November 30, 2010.

Seung-Min Jung is with the Department of Information Science and Telecommunication, Hanshin University, Osan, 447-791, Korea (Tel: +82-31-379-0661, Mobile: +82-19-286-2571, Email: jasmin@hs.ac.kr)

Based on the minutiae, the input fingerprint is compared with the enrolled fingerprint. To match two fingerprints captured with unknown direction and position, the differences of direction and position between the two fingerprints are detected and aligned. This alignment stage estimates transformations, such as translation and rotation, between the two fingerprints and aligns two minutiae according to the estimated parameters. If the alignment is accurate, the following matching stage is a simple point pattern matching. The matching stage compares two minutiae based on their position, orientation, and type. A matching score is then computed. Fig. 4 shows the Matching Algorithm

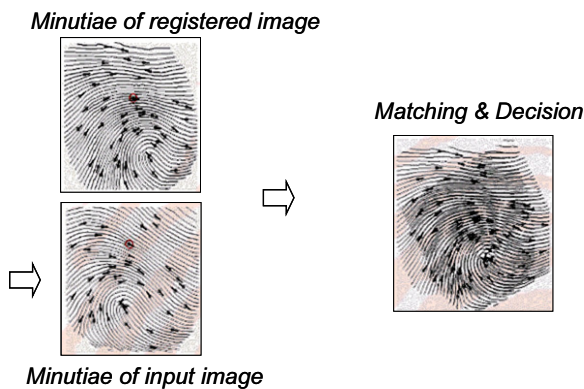


Fig. 4 Matching Algorithm

Conventional fingerprint authentication hardware and software system needs a high performance microcontroller such as 32bit ARM7, or ARM9 CPU for completing the processing time below 1 second[3][6]. Fig. 5 shows a conventional fingerprint authentication hardware and software system.

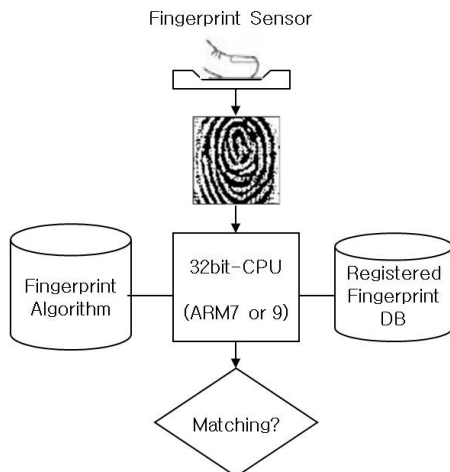


Fig. 5 Typical fingerprint identification system

This paper proposes an effective hardware scheme for thinning stage processing of a fingerprint identification

based on minutiae from analyzing of algorithm cycle distribution. The cycle distribution of each algorithm step is analyzed in FPGA environment. Hardware block processing is more effective than software one in speed, because a thinning algorithm is iteration of simple instructions. The ZS algorithm[4] is applied for a thinning stage. The hardware scheme is designed and simulated in RTL. The logic was also synthesized by Synopsys Design Compiler using 0.35 $\mu$ m CMOS process. The layout is performed by an auto placement and routing tool.

## II. ANALYSIS OF ALGORITHM

We implemented general purpose ARM7 compatible RISC microcontroller by FPGA to verify proper operation of the proposed fingerprint algorithm. Fig. 6 shows the FPGA emulation board for verifying the algorithm.

The design was composed of 3,336 slices(25%), 2,469 registers(9%), and 4,613 LUT(18%) with FPGA-XCV800Q240. The total equivalent gate count was 49,500 and the operating speed was 40MHz. We inserted the checkpoint routines at each step of the identification algorithm. The checkpoint routines successively turn on one of 8 LEDs on the emulation board. We finally confirmed that our embedded 32-bit RISC core successfully carried out the fingerprint authentication algorithm. The FPGA emulation board has 800K gates XCV and external memory devices.

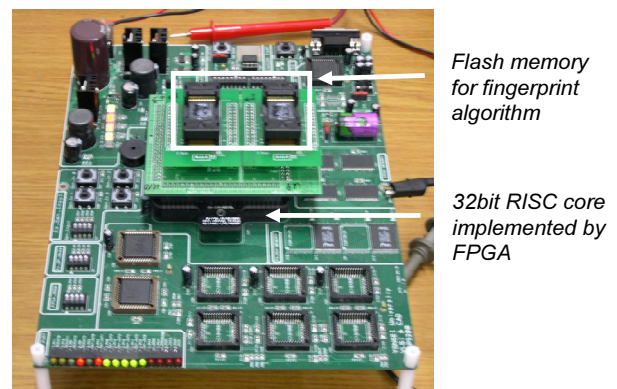


Fig. 6. FPGA emulation board for verifying the algorithm

All the results of algorithm instructions are compared with those of ARMulator, which is a software emulator of the ARM processor, at every algorithm cycle as shown in Table 1 and Fig. 7. The total clock number of our algorithm was 44.7M cycles and the operating speed was 1.1sec at 40MHz with a 160x192 sample image. GABOR filtering and thinning step occupied about 80% of total cycle. Minutiae detection time including matching is less than 3%. The

thinning step occupies 39%(17.8M cycles). The processing time of thinning step is 0.446 second at 40Mhz. Table 1 and 2 show the results of ARMulator simulation. Here, QC : Quality Check, BO : Block Orientation, SBO : Smooth Block Orientation, RF : Ridge Frequency, FGF : Frequency Gaussian Filter, FLPF : Frequency LowPass Filter, GF : Gabor Filter T : Thinning, MD : Minutia Detection, DFM : Detect False Minutia, RM : Read Minutiae from a file, MP : Matching Processing

TABLE 1.  
INSTRUCTION CYCLE DISTRIBUTION

stage	QC	BO	SBO	RF	FGF	FLPF
instruction	215059	1,375,079	253,535	2,918,970	13,029	107,652
S-cycles	245781	1,414,353	304,618	2,873,557	14,461	114,166
N-cycles	61446	366,037	99,830	847,425	3,174	14,407
I-cycles	30721	429,333	34,733	487,788	2,365	5,023
C-cycles	0	0	0	0	0	0
Total	337,948	2,209,723	439,181	4,208,770	20,000	133,596
%	0.75%	4.91%	0.98%	9.35%	0.04%	0.30%
clock	Total processing time(sec)					
40Mhz	0.008	0.055	0.011	0.105	0.001	0.003

TABLE 2.  
RESULTS OF ARMULATOR ANALYSIS(CONT.)

stage	GF	T	MD	DFM	MP	Total clocks
instruction	12,496,058	9,548,256	552,487	443,587	81,920	27,790,573
S-cycles	12,838,764	9,602,337	667,456	489,315	112,700	28,431,727
N-cycles	3,213,533	5,635,665	197,734	200,131	3,797	10,581,733
I-cycles	2,250,996	2,597,056	71,698	100,584	7,972	5,987,548
C-cycles	0	0	0	0	0	0
Total	18,303,293	17,835,058	936,888	790,030	124,469	45,001,008
%	40.67%	39.63%	2.08%	1.76%	0.28%	100.00%
clock	Total processing time(sec)					
40Mhz	0.458	0.446	0.023	0.020	0.003	1.125

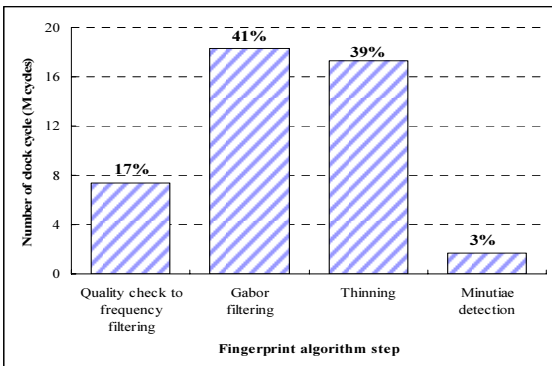


Fig. 7 Instruction distribution graph

The thinning step is needed to be processed by hardware block, because it is performed repeatedly by processing the same operation using an image window masking method. It can reduce the burden of the system and improve speed.

### III. ASIC IMPLEMENTATION OF THINNING STAGE

The well-proved thinning algorithms are ZS(Zhang and Suen)[5]. ZS parallel thinning algorithm performs sub-iteration step twice in 3x3 image pixel window as shown in Fig. 8 and Fig. 9.

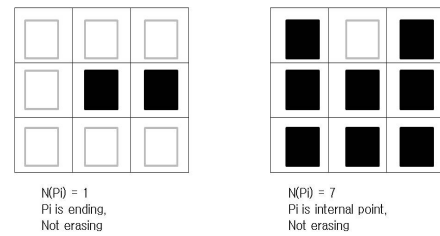


Fig. 8 Number of black pixel neighbor P(i)

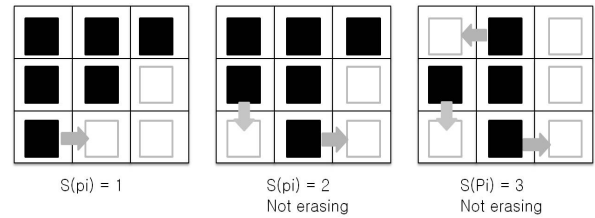


Fig. 9 Number of black to white change neighbor P(i)

The first step:

The pixels satisfied with following conditions are erased.

1.  $2 \leq N(P_i) \leq 6$
2.  $S(P_i) = 1$
3.  $P_2 * P_6 * P_8 = 0$
4.  $P_4 * P_6 * P_8 = 0$

Here,  $N(P_i)$  is the number of value 1 in 8-neighbor pixels of Fig. 6 and expressed in following equation (1).

$$N(P_i) = P_1 + P_2 + \dots + P_8 \quad (1)$$

And, as shown in Fig. 7,  $S(P_i)$  means the number of 1 to 0 ( $1 \rightarrow 0$ ) patterns in 8-neighbor pixels.

The second step:

Condition 3 and 4 in the first step are replaced with the following conditions.

- 3'.  $P2 * P4 * P8 = 0$
- 4'.  $P2 * P4 * P6 = 0$

Hardware block processing is more effective than software in speed, because a thinning algorithm is iteration of simple instructions. This paper describes an effective hardware scheme for thinning stage processing using the Verilog-HDL in 160x192 Pixel Array as shown in Fig. 10. It replaces 32bit CPU by 16 or 8bit CPU.

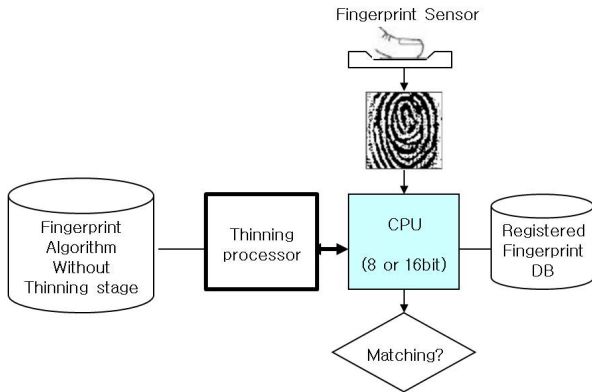


Fig. 10 Proposed fingerprint system

Iterations of 160x192 pixel array are performed for ZS algorithm in FPGA environment to verify proper operation. Fig. 11 shows the proposed thinning processor block diagram. The 160x192 counter generates 2 dimensional address for selecting fingerprint pixel array. The design was composed of 113,539 slices, 94,640 flip-flops, and 125,360 LUT with FPGA-XC4VLX200 including 2 memories of 31Kbits. Synthesis report shows the thinning processor can be operated in minimum 14.3ns clock period. Maximum operating speed was 70MHz. The processing time of thinning step 1 and 2 is 16.2ms at 40MHz by total 65000 clock cycles. It means a thinning processing time can be reduced from about 400ms of a conventional system to 16.2ms, dramatically. The 3x3 pixel window gen block generates 8-neighbor pixels of  $P(i,j)$  for 3x3 window. The thinning stag1 and 2 blocks process first and second steps of ZS algorithm, respectively.

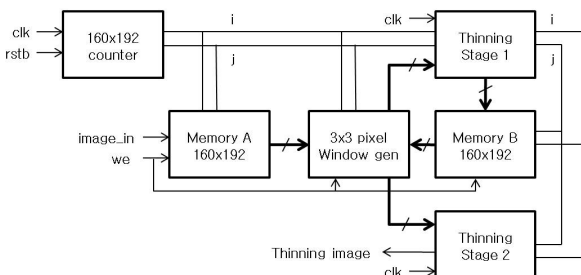


Fig. 11 Block diagram of thinning processor

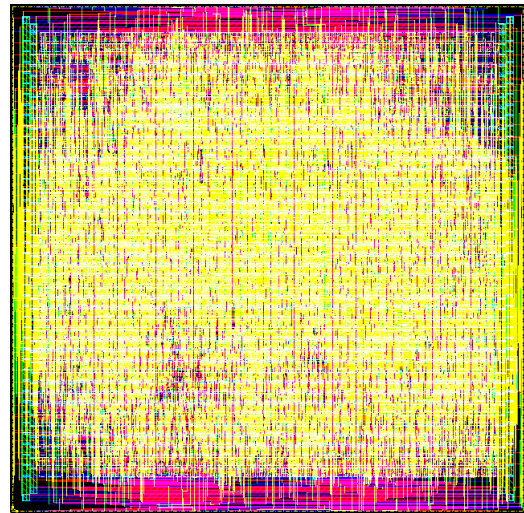


Fig. 12 Layout of thinning processor (0.35µm CMOS process, 4-metal)

Fig. 12 shows the layout of thinning processor in 0.35µm CMOS process with 4-metal using an auto placement and routing tool. Area is 739µm x 732µm (0.541 mm<sup>2</sup>) and gate count is 10,222 without memory. Therefore, the burden of chip area is small relatively in respect of 39% speed improvement. To confirm the operation of the proposed thinning processor block, we extracted each parasitic capacitance from the optimized layout and performed the post-simulation on condition of 0.35µm worst. Fig. 13 shows the thinned image of 160x192 pixel sample. The left image is original sensor output data and right is simulation output of thinning processor. The result shows the proposed thinning processor can successfully perform a fingerprint verification algorithm in ASIC. We expect maximum 39% improvement of algorithm speed, replacement 32bit CPU by 16 or 8bit CPU, low power consumption and small size fingerprint system from the result.



Fig. 13 160x192 sample image post-simulation result

#### IV. CONCLUSIONS

In this paper, we propose an effective hardware scheme for thinning stage processing of a fingerprint identification based on minutiae from analyzing of algorithm instruction cycle distribution. There are several steps in the verification process. Each step of a fingerprint algorithm is analyzed based on FPGA and ARMulator. A thinning stage occupies 39% cycle of 32-bit RISC microprocessor system for a fingerprint identification algorithm. Hardware block processing is more effective than software one in speed, because a thinning algorithm is iteration of simple instructions. The ZS algorithm was applied for a thinning stage. The hardware scheme was designed and simulated in RTL. The logic was also synthesized. The layout was performed based on an auto placement and routing in Magna 0.35 $\mu\text{m}$  CMOS process. Area was 739 $\mu\text{m}$  x 732 $\mu\text{m}$  (0.541  $\mu\text{m}^2$ ) and gate count was 10,222 without memory. The post-simulation result shows the proposed thinning processor can successfully perform a fingerprint verification algorithm in ASIC. We expect maximum 39% improvement of algorithm speed, replacement 32bit CPU by 16 or 8bit CPU, low power consumption and small size fingerprint system from the result.

#### ACKNOWLEDGMENT

This work was supported by Hanshin University Research Grant.

#### REFERENCES

- [1] Lin Hong, Yifei Wan, and Anil Jain, "Fingerprint Image Enhancement: Algorithm and Performance Evaluation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 8, pp. 777-789, August 1998.
- [2] Chul-Heui Lee, Min-Seob Lee, "An Efficient Fingerprint Image Classification using Singular Points and Gabor filter," Journal of Telecommunications and Information, Vol. 7, 2003.
- [3] S.M. Jung, J.M. Nam, D.H. Yang, and M.-K Lee "A CMOS Integrated Capacitive Fingerprint Sensor With 32-bit RISC Microcontroller," IEEE Journal of Solid-State Circuits, vol. 40, no. 8, pp. 1745-1750, Aug. 2005.
- [4] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns", Communications of the ACM, vol. 27, no. 6, pp. 236-239, Mar. 1984.
- [5] Jun-Sik Kwon, "Obtaining 1-pixel Width Line Using an Enhanced Parallel Thinning Algorithm," Journal of IEK, Vol. 46- SP., No. 1, January 2009.
- [6] S.M. Jung, "Thinning Processor for 160x192 Pixel Array Fingerprint Recognition," International Journal of KIMICS, Vol. 8, No. 5, 571-576, Oct. 2010.

**Seung-Min Jung** (M'02) Refer the (IJMICS) International Journal of Maritime Information and Communication Sciences, Vol. 6, No. 1, 2008. He is associate professor in Department of Information Science and Telecommunication, Hanshin University, Osan, Korea.