Volume 3, Issue 8, August 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Multi Value Logic Based - Finger Print Unit

A Devendran
Research Scholar – St Peter's University
Faculty – Dr MGR Educational & Research Institute
India

Dr Bhuvaneshwari T
Professor
Loganatha Narayanasamy Government College
India

Dr Uma Maheswari J L
Associate Professor
Dr MGR Educational & Research Institute
India

K Arun Kumar
Technical Architect
PplInfotech Sol Pvt Ltd.
India

Abstract*— We propose a new application processing unit (Coprocessor) for Finger Print recognition based on Multi-valued Logic. We created a simulator for Multi-valued Logic - modeling Memory, Cache, Finger Print Unit with multiple FP sub-units. Experiments were made on radix-4 and radix-8 system and compared with Binary counterpart on a similar setup.

Keywords*— IFinger print, Multi-valued Logic,

I. INTRODUCTION

Multiple-valued logic (MVL) - possibility to represent the information with more than two discrete levels. Representing data in a MVL system is more effective than the binary based representation, because the number of interconnections can be significantly reduced. The real advantage of higher-radix arithmetic is fewer arithmetic operations. Especially in bulk signal processing units, they may bring in highly sophisticated but less complicated expressions.

Why Finger Print Application? - Fingerprint identification is a popular biometric identification technology. It includes fingerprint verification and fingerprint recognition. Both of them have high computations on based on complex formulas. They use minutiae, such as end points and bifurcation points, as features. To correctly extract minutiae from fingerprint images becomes an essential step in fingerprint identification. It involves lot of computation for all its operations, makes it an ideal candidate for Multi-valued logic application.

II. BACKGROUND & MOTIVATION

There has been lot of research going in multi-valued logic and processing units. Some of our observations are listed below with referenes.1

Characteristics of radix-8 MVL and binary adders [1]

Table 3. Performance comparison of new MVL-based and binary adders (T = time duration of one step).

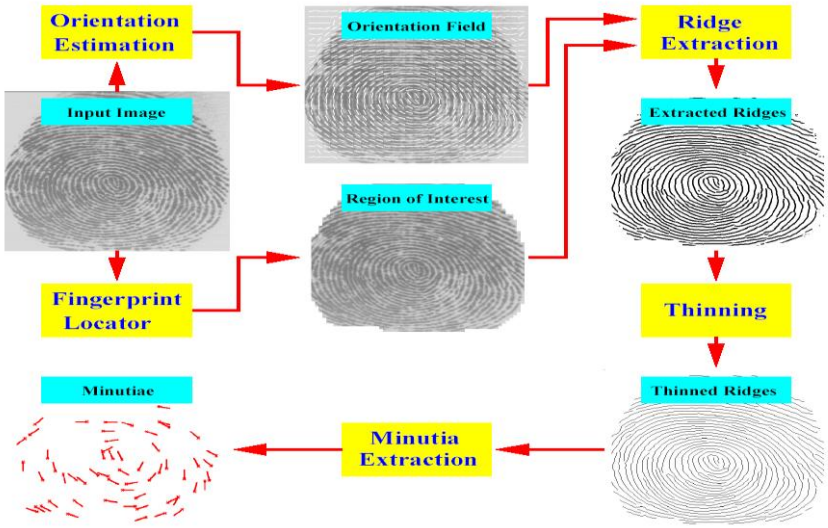
Adder	Trans. count	Max. delay (ns)	Av. power diss. (mW) $T = 40$ ns	Av. power diss. (mW) $T = 10$ ns	Area $(\mu m)^2$
MVL, radix-8	21	3.2	0.95	1.05	87 × 24
3-bit RCA	84	6.5	0.14	0.52	160 × 85
3-bit CLA	108	7.3	0.23	0.93	175 × 93

Higher radix floating-point Implementations - Higher radix floating-point representations have been in use for many years, as mentioned earlier, although they are not very common at present. The design of a native hexadecimal FPU which also operates on binary, IEEE operands is described in [8].

Memories - (DRAM / Cache) -In memory technology, recent applications of multiple-valued logic include Flash [3], DRAM [4], and optical [5] memory designs. An overview of CMOS-related multiple-valued memory technologies can be found in [4].

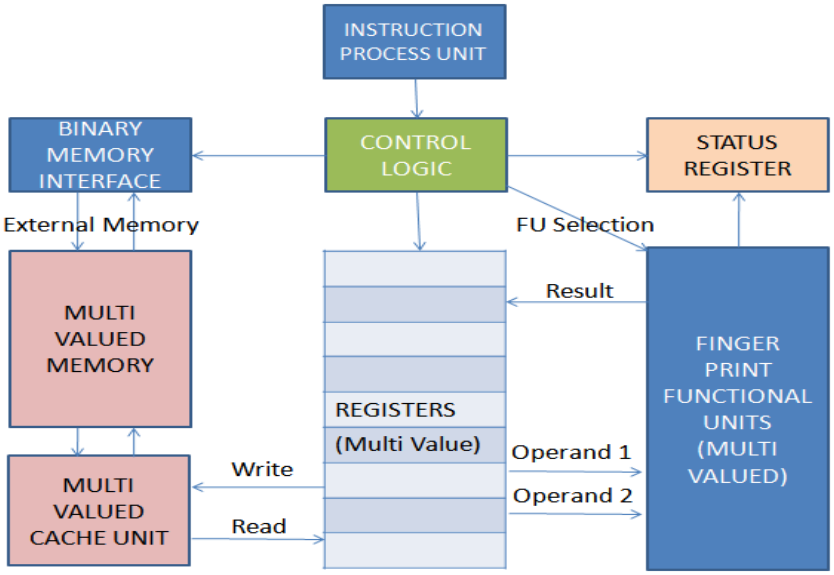
Arithmetic circuits - Multipliers/Adders - The most convincing demonstration of the successful application of multiple-valued logic to the design of multipliers is 200 MHz 54 X 54-bit multiplier designed using multiple-valued current-mode MOS circuits [9]. The performance of the multiplier is evaluated to be about 1.4 times faster than that of a corresponding binary implementation under the normalize power dissipation. Reference[2]

Image process – Finger point recognition - Fingerprint Matching Techniques - Minutiae-based, Correlation-based,



Ridge Feature-based.

III. FLOATING POINT UNIT – SIMULATOR IMPLEMENTATION



Registers – Our system has 2 set of Radix-8 based registers. (Capable of loading entire FP Minutia details of an entire finger print in a set)

On-Chip Memory – Our system is proposed with a on-chip memory of multi-valued logic to store 500 finger print minutia (FP templates). This 500 is the working set of finger prints for the Coprocessor. i.e. all its operations access & manipulating them. Our system gives user/programmer the flexibility to load new Finger Prints (processed ones) can be brought in from system file system by giving the fully qualified path – by swapping (LRU replacement Algorithm).

Cache – A Fully Associative cache (with LRU replacement Algorithm) is implemented for a faster access to memory. It caches 512 blocks recently used and address of the block is used as the tags for comparison

Finger Print Unit

Load Unit – Load FP details (Minutia array) for operation into one of the set of registers

Store Unit – Store the FP details (Minutia array) back in memory

Compare Unit - Compare the FP details in register sets and updates the status register with the matching point count. Most of the functional unit impelmentations are complex – implementation details is out of scope for this report

Orientation Field Extractor – Finds angle formed by the ridges with horizontal axis
FP Area Locator - Separates fingerprint area from the background
Ridge Extractor – Extarcts Ridges and furrows with filters, as they run parallel to one another forming 2-D sine wave
Thinner - Thinning creates a 8-connected thinned ridges of one pixel width
Minutia Extractor - Spikes and breaks in the thinned ridge map leads to detection of spurious minutiae. Heuristics are applied to post-process the thinned ridge map. If the angle between the branch and the trunk is greater than 70ºand less than 110º, then the branch is removed. If the break in the ridges is less than 15 pixels and no other pixel passes through it, then the break is connected to avoid errors.

Instruction Processing Unit

Our system has an Instruction processing unit, which has a Instruction queue. This queue holds 'users' or 'developers' programs done for our coprocessor – which can be configured to be loaded when the system starts. Following section explains the details of the instruction set. IPU processes instructions from the queue one by one.

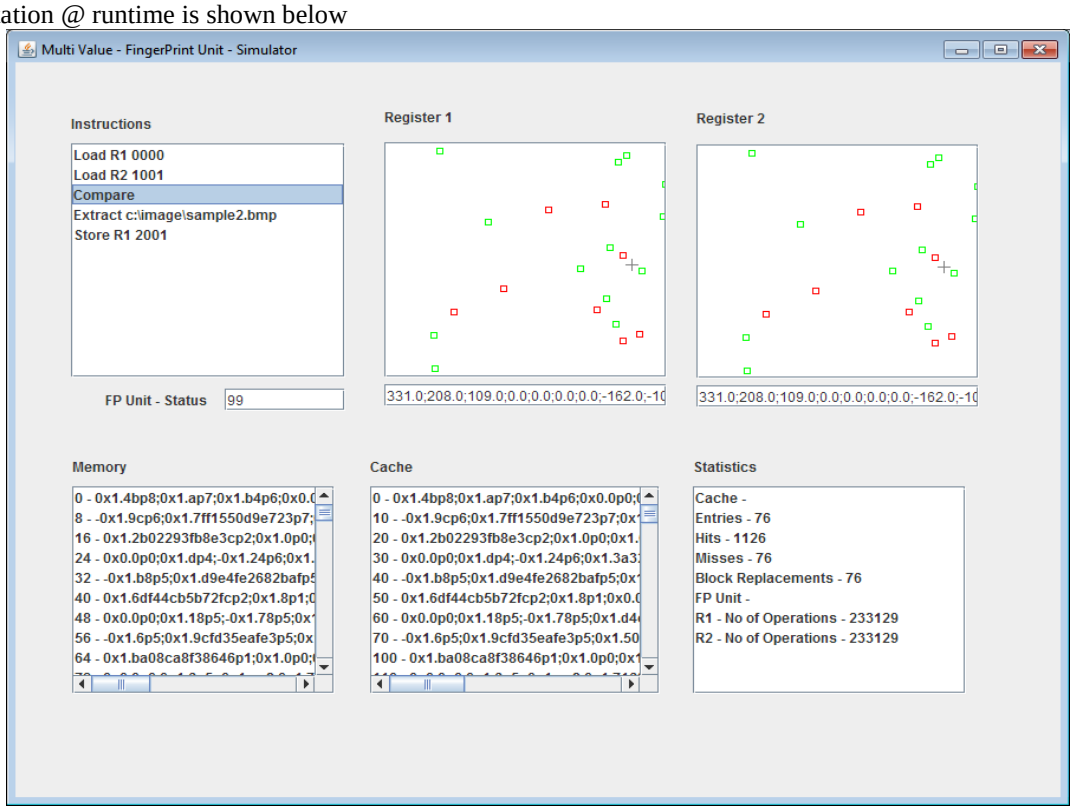
1) Instruction set:

Load R1 0000 – Loads finger print details (minutia set) starting @ memory 0000 into R1 register
Store R1 1001 – Store finger print details (minutia set) R1 register into memory starting @ 1001
Compare – Compare finger prints in R1 & R2 and updates the matching count in status register
Extract c:\images\sample.bmp – Extract the minutia details from bmp file (fully qualified path mentioned in the instruction) and result is available in R1 register

IV. EXPERIMENT & RESULTS:

Experiment Setup:

We have implemented a Multi valued Finger Print Simulator with the above mentioned unit and a screenshot of our implementation @ runtime is shown below



We compared multi value logic performance against corresponding binary counterpart (radix-8, radix-4, radix-2) for the following operations

Memory related operations – Load, Store
Computation operation – Compare

For calculating the matching point our implementation does 2 trignometric operation, 6 multiplication operation, 4 addition operations & 2 round-off operation as per the below code (+some comparisons).

1 Compare operation = 2 Trignometric operation+ 6 multiplication operation + 4 addition operations + 2 absolute operation per the code below

```
// do angle shift for x
x1 = r * Math.cos(d + (k * radian));
resx = Math.abs((int) x2 + (int) (-1 * x1));
// do angle shift for y
y1 = r * Math.sin(d + (k * radian));
resy = Math.abs((int) y2 + (int) (-1 * y1));
```


Since trignometric functions themselves are complex. For our experiment we assumed it uses 10 multiplication operations internally. (Actually it is an exponential series involving squares & factorials). Also, we assumed absolute will have similar time of an addition.

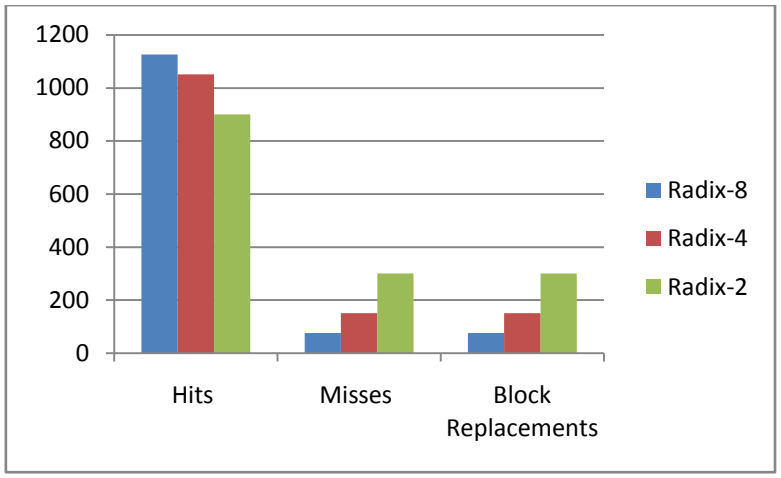
Hence, in our experiment we assumed for analysis

1 compare = 24 multiplication + 6 addition

Results

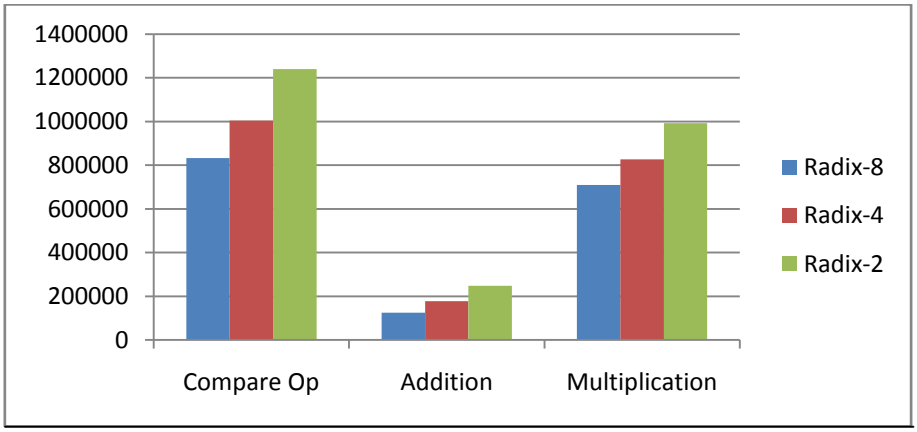
Memory Operations: (2 Load Operations)

	Hits	Misses	Block Replacements
Radix-8	1126	76	76
Radix-4	1051	151	151
Radix-2	901	301	301



Computations: Compare Operation

	Compare Op (No. of Cycles)	Addition(No. of Cycles)	Multiplication(No. of Cycles)
Radix-8 (2 - scaled from base 2)[1]	620235	124047	496188
Radix-4 (1.4 scaled from base 2)[2]	886050	177210	708840
Radix-2	1240470	248094	992376



V. CONCLUSION & FUTURE EXTENSION:

1. Comparing the results we clearly have better processing time and the cache hits resulting in better performance again - with the multi-valued systems.
2. Having a separate unit for specific application – might helps us to decouple from binary world into multi valued arena.

3. The simulator is built from scratch in Java technology – which simulates a basic memory, cache, process flow. This can be taken to model/extend another functionality.
4. Our study is not complete in terms of accurate results and one would naturally wants to do it.
5. Other Image/Signal processing Applications - can be studied under similar conditions.

References:

[1] Implementation of Multi-Valued Logic Gates Using Full Current-Mode CMOS Circuits - TURGAY TEMEL∗ AND AVNI
[2] Multiple-Valued Logic in VLSI: Challenges and Opportunities - Elena Dubrova
[3] K. Takeuchi et al., A multipage cell architecture for high-speed programming multilevel NAND flash memories, *IEEE J. Solid-State Circuits* **33**, 8, (1998), 1228-1238
[4] B. Ricco et al., Non-volatile multilevel memories for digital applications, *Proc. IEEE* **86**, 12, (1998), 2399-2421.
[5] H. Kimura, T. Takahira, New concept for multiple-valued optical memory, *Electronics Letters* **33**, 10, (1997), 847-848
[6] Minutia vs. Pattern Based Fingerprint Templates – White paper
[7] Fingerprint Recognition - Anil K. Jain / Michigan State University
[8] E. M. Schwarz, R. M. Smith, and C. A. Krygowski, "The S/390 G5 Floating Point Unit Supporting Hex and Binary Architecture," Proceedings of the 14th IEEE Symposium on Computer Arithmetic, 1999

