Associative Memory Based Hardware Design for An OCR System and Prototyping with FPGA

Ali Ahmadi, M.D., Anwarul Abedin, Kazuhiro Kamimura, Yoshinori Shirakawa, Hans Jürgen Mattausch, and Tetsushi Koide

Research Center for Nanodevices Systems, Hiroshima University, 1-4-2 Kagamiyama, Higashi-Hiroshima, 739-8527, Japan

1. Introduction

Much progress has been already made over the past years on development of hardware for optical character recognition (OCR) systems [1] and different movable OCR products are presently in the market [2] but yet they hardly ever afford the desired robustness and hardware size, simultaneously. In this research we propose an associative memory based OCR system for real-time character recognition implemented in an FPGA architecture. The prototype of associative memory we use here as the main classifier is already designed in our lab [3] and has a mixed analog-digital fully-parallel architecture for nearest Hamming/Mannhattan-distance search. In this work, to have a fast prototyping the OCR preprocessing steps as well as the associative memory unit are mapped on the FPGA and the system performance is evaluated with some real data samples.

2. OCR Processing Units

The main steps considered for OCR process are: data reading, binarizing, noise removal, image labeling, segmentation, and classification. In data reading step the data of each text line are scanned continuously as a sequence of thin frames by moving a reading device (scanner sensor) on the text. The frames between each two word spaces are collected and form a larger frame as a gray-scale bitmap array which contains all the word characters. In the binarizing step this frame is binarized to a simple black-white bitmap by taking a local threshold value extracted via a mean filter. In order to remove noise from the frame, a median filter with neighborhood of 2×2 is applied. Next, by employing a sequential labeling algorithm different segments of the image frame are labeled and segments larger than a threshold level are recognized as a single character. Further detail about labeling algorithm is given in the next Section.

To have an accurate classification the size of each character is normalized to 16×16 pixels before classification. We use a bilinear interpolation algorithm for resizing the character bitmap. The last and main step of the process is character classification which is carried out by a nearest-distance search algorithm applying the associative memory. The normalized segmented character is matched as a 256 bits vector to a number of reference patterns using the Manhattan distance measure and the reference pattern with minimum distance is considered as the winner class. The design is described in Verilog-HDL and synthesized using the Synplify-pro compiler and then implemented in the Altera FPGA family using the QuartusII tool for placement and routing. Using Stratix DSP development kit EP1S80 and a clock frequency of 50 MHz, a total number of 2,337 logic cells and 36 Kbits of SRAM memory are used for placement and routing of associative memory part. The timing simulation results are reported in the next Section.

4. Analysis of Simulation Results

We examined the system with some real data samples of English text characters and evaluated the results with a software program. A total number of 16 data set including different fonts (Times and Arial), noisy data, color background data, slightly rotated data, and data with different resolution were gathered and tested. Each set contained 26 characters and as mentioned before, each data sample is considered as a 256 bits vector. The experimental results of distance-matching between data vectors and reference patterns showed overall of 2 misclassification cases (0.5%) for noisy data and zero case for other data type. The mini-
mum distance between winner and nearest-loser over all the data samples is averagely 2.5 which is not yet reliable enough. The minimum distance of winner and second loser is 13 bits.

Figure 5(a) indicates the winner-input distance for different data samples. The average winner-input distance for all the input samples was calculated and found as 31 bits. Having this distance and referring to plot of Fig. 5(b) which gives the typical winner search time of the associative memory according to winner-input distance, we can find the average search time of 128 ns for classification of each test sample. This is the search time within 128 reference patterns and will be changed in case of increase in reference patterns number. Comparing to other OCR products existing in the market, however this prototype model is not still robust enough but is advantageous in terms of classification time and hardware size. We are also planning to develop the system with a learning algorithm for optimizing the reference patterns selection process.

References

---

Table 1: Characteristic data of designed associative memory test chip.

<table>
<thead>
<tr>
<th>Distance Measure</th>
<th>Hamming</th>
<th>Manhattan (5 bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Field</td>
<td>32 x 768</td>
<td>128 x 80</td>
</tr>
<tr>
<td>Technology</td>
<td>0.6 µm CMOS</td>
<td>0.35 µm CMOS</td>
</tr>
<tr>
<td>Area</td>
<td>9.11 mm²</td>
<td>8.6 mm²</td>
</tr>
<tr>
<td>Search Range</td>
<td>0 - 400 bit</td>
<td>0 - 480 bit</td>
</tr>
<tr>
<td>Winner-Search Time</td>
<td>&lt; 70 nsec</td>
<td>&lt; 190 nsec</td>
</tr>
<tr>
<td>Performance</td>
<td>1.34 TOPS</td>
<td>160 GOPS</td>
</tr>
<tr>
<td>Power Dissipation</td>
<td>43 mW</td>
<td>91 mW</td>
</tr>
<tr>
<td>Supply Voltage</td>
<td>3.3V</td>
<td>3.3V</td>
</tr>
</tbody>
</table>

---

Fig. 1. Associative memory architecture.

Fig. 2. The schematic of the binarizing block using local threshold algorithm implemented in an FPGA architecture.

Fig. 3. Flowchart of segmentation procedure. N is number of distinct labels (Li, i=1…N). SLB1, SLB2 are buffers for recording equivalent labels, SM is a temporary memory for saving addresses of pixels with label Li, and cmin, cmax, rmin, rmax are min. row, max. row, min. column, max. column of segment Li, respectively.

Fig. 4. The schematic of associative memory mapped with 4 dual-port RAMs in an FPGA architecture. 16 matchings are performed within each clock cycle.

Fig. 5 (a). Winner-Input distance for different data of font Times. (b) Average winner search times in associative memory.
Associative Memory Based Hardware Design for An OCR System and Prototyping with FPGA

A. Ahmadi, M.A. Abedin, K. Kamimura, Y. Shirakawa, H.J.Mattausch, and T. Koide
Research Center for Nanodevices and Systems, Hiroshima University

Background & Research Objective

Optical character recognition (OCR) systems have been widely used in recent years and various approaches are applied for developing their hardware and processing algorithms.

As for a small mobile OCR, e.g. a cognitive pen, an ideal model is thought as a system with high accuracy and speed, and minimum hardware size & power dissipation at the same time.

We propose an associative memory based OCR system for real-time character recognition implemented in an FPGA architecture.

An associative memory-based OCR with qualifications:

1. Recognition of printed characters & words
2. Robustness to Noise, Rotation, Color
3. Applicable to different Fonts
4. High Speed
5. Learning capability

Processing Steps

Data Reading

Real-time data reading problems:
- Rotation & shift
- High rate of noise
- High speed of recognition needed

Local thresholding

Thresholding image

Noise Removal

Character separation

Size normalization

Classification

(Associative memory with nearest-match Manhattan distance)

Flowchart of data segmentation (after labeling).

Hardware Implementation

Preprocessing steps are designed and implemented in an FPGA architecture. We use Altera Stratix DSP development kit EP1S80 with a clock of 50 MHz as the main platform.

The schematic of binarizing block using local threshold algorithm implemented in an FPGA architecture. The core part of algorithm can be generalized for implementation of any FIR filter.

The prototype of associative memory we use here as the main classifier is already designed in our lab as an LSI chip and has a mixed analog-digital fully-parallel architecture for nearest Hamming/Manhattan-distance search.

Conclusions

1. The proposed associative memory based OCR is advantageous in terms of classification speed and hardware size.

2. Due to fully parallel pattern-matching used in the associative memory the average search time for each character is obtained as 129 ns which is very much faster than other existing OCRs.

3. We are planning to equip the system with a learning algorithm which updates and optimizes reference patterns continuously over the time using two types of short term and long term memory.