

A VLSI ARCHITECTURE FOR MANHATTAN DISTANCES AND WTA

Huynh Huu Thuan, Le Duc Hung

Department of Electronics - Telecommunications
University of Natural Sciences, Vietnam National University – HCM

ABSTRACT

A VLSI architecture for calculating Manhattan distances and finding the minimum value (Winner Take All - WTA) has been designed to aim at image recognition and compression (using vector quantization - VQ) applications. This architecture uses SIMD and two stages pipeline. The result is tested on FPGA Stratix and its configuration can be changed easily. We do not develop any image processing theory but just design the VLSI architecture for real time processing based on hardware.

1. INTRODUCTION

There has been a considerable interest in the development of image processing systems in real-time by VLSI chips, and many algorithms have been developed for this. In many image recognition systems, features of the image are extracted and form feature vectors. These feature vectors are stored in the memory on chip and are used to compare with an input vector to find the maximum-likelihood vector (image matching) using pattern matching technique [1], [2]. In image compression using VQ [3], [4], a macro block of pixels is approximated by the best matched pattern in the codebook, where typical patterns are stores as templates, and its code is transmitted. The decompression is carried out by pasting the corresponding pattern at the proper location.

In this paper, we design a VLSI architecture for pattern matching in Fig. 1.

Before this system operates, template vectors (T_1, T_2, \dots, T_n) are downloaded from outside (by a personal computer, a DSP or so on...). When all template vectors are downloaded, an input vector (X) is given to the system (using SRAM or FIFO) and the output is the code of the maximum-likelihood vector. Absolute Value and Accumulate (AVA) blocks are used to calculate the Manhattan distances d_j between the template vector T_j and the input vector X by

the expression $d_j = \sum_{i=1}^m |t_{i,j} - x_i|$ where j is the

code number of the T_j , $t_{i,j}$ is the i th element of T_j , x_i is the i th element of X and m is the number of elements of each vector. The WTA circuit is used to search the template vector having the minimum distance to X . The Winner Observer (WO) circuit is used to encode the location of the winner and if there are more than two winners this circuit will select only one of them.

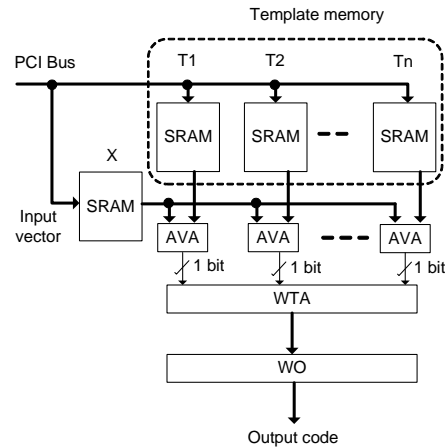


Fig.1. Block diagram of design used for pattern matching.

The result is tested on FPGA Stratix and PPED vectors [2] are used as testing vectors.

2. CIRCUIT DESIGN

Fig. 2 shows the block diagram for downloading data to template vectors, the X vector and generating addresses.

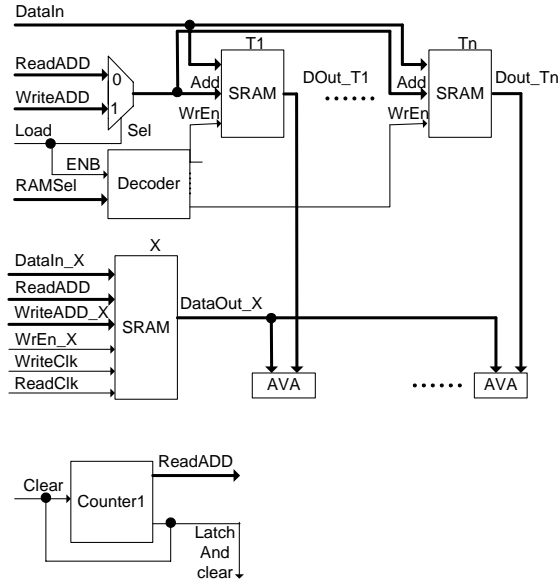


Fig. 2. The block diagram for downloading data and generating addresses.

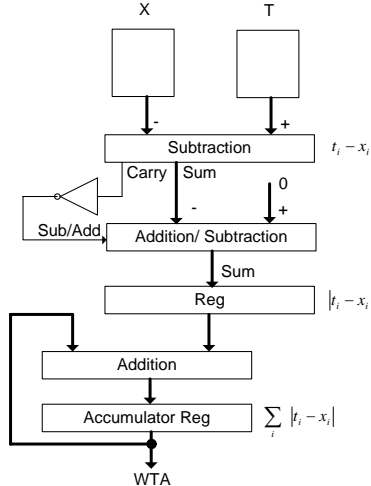


Fig. 3. The AVA circuit for calculating the Manhattan distance.

When the Load signal is 1, one of template vectors is selected by the Decoder and data for this vector is downloaded by DataIn, WriteADD signals. The X vector is stored in a dual-port RAM (or FIFO) and downloaded by DataIn_X, WriteADD and WrEn_X signals. When the Load signal is 0, all template vectors and the X vector are simultaneously read out from element to element by the address generated by the counter1. This counter also creates the 'Latch and Clear' signal for pipeline operation and to reset the counter when the last element is read out. Fig. 3 shows

the AVA circuit for calculating the Manhattan distance between the X vector and a template vector in an element-serial manner.

The first, the element of the template vector is subtracted from the element of the X vector and then the carry bit is used to correct the result to a corresponding positive value. As a result, the Manhattan signal is stored in the Accumulator register. When the last element of the template and the X vector is calculated and accumulated, the Accumulator register is latched into the WTA circuit by the 'Latch and Clear' signal to do the second pipeline stage.

Fig. 4 shows the WTA circuit to find the minimum Manhattan distance. The distance values from AVA circuits are latched into shift registers. A register contains status flags to indicate the status of the competition: 1 means the winner and 0 means the loser. At the beginning, all flags are set to 1 and then distance values from shift registers are compared in order from MSB to LSB in one clock. At each clock, the flags of inputs having larger values will be 0 and withdrawn from the competition. When the last bit (LSB) is fed into the circuit, the counter2 will generate a signal to latch the result.

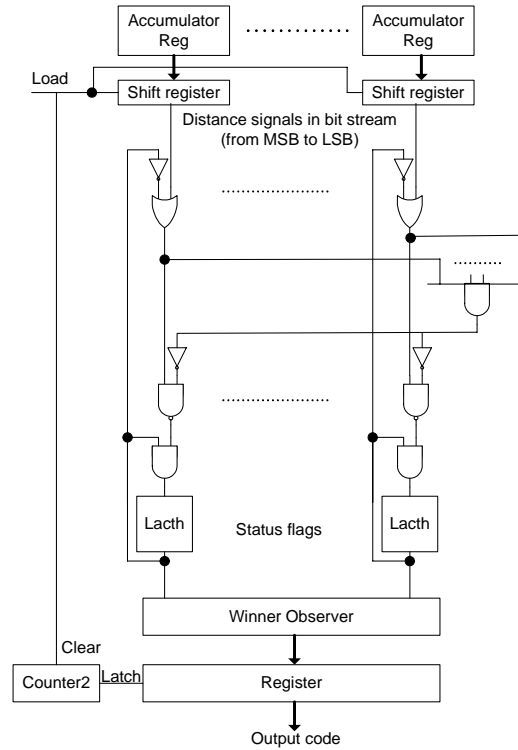


Fig. 4. The WTA circuit.

The Winner Observer is a priority encoder used to encode the position of the winner and if there are more than two winners (more than two values 1 in status flags) then this circuit will select only the smaller position. Based on the function of the counter1 and counter2, we can design a circuit that the number of elements of vectors can be changed for different applications.

3. EXPERIMENTAL RESULTS

The result is tested by Projected Principal Edge Distribution - PPED vectors [2]. For the testing purpose, we just use a 32-dimension vector for the original 64x64 image by series connecting set of components in the order of horizontal and vertical instead of a 64 - dimension vector in the original paper [2]. Eight Vietnamese letters and their PPED vectors are shown in Fig. 5. These vectors are downloaded to template vectors in SRAM blocks and the X vector is one of them. Because there are 32 elements for each vector, it takes 32 clocks to obtain the Manhattan distance, one clock for reading data from SRAM blocks, two clocks for calculations and data transfer to the accumulator register. As a result, we can obtain the Manhattan distance after 35 clocks.

Because the accumulator stores the sum of 32 elements and 8 bits/element, its size is 13 bits. This means that the WTA circuit takes 13 clocks to obtain the result.

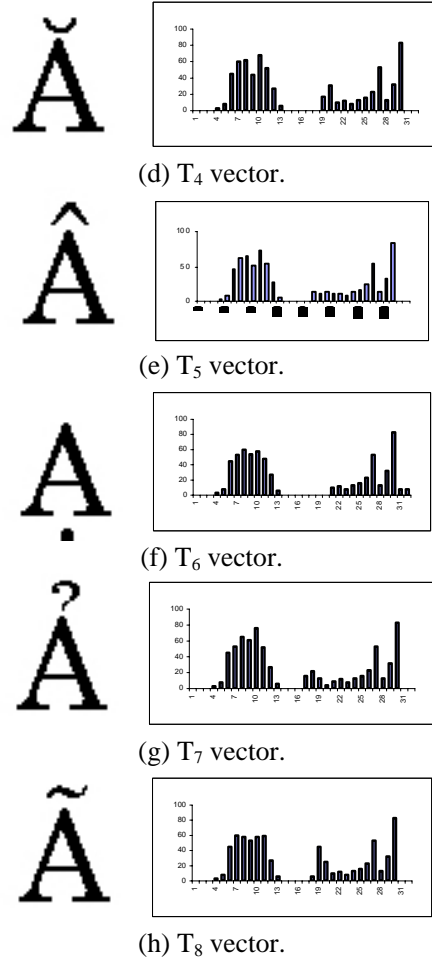
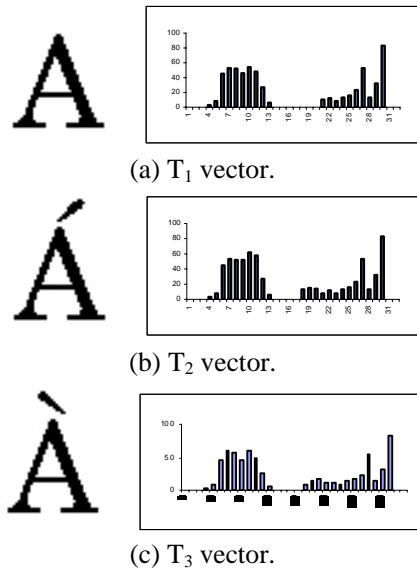


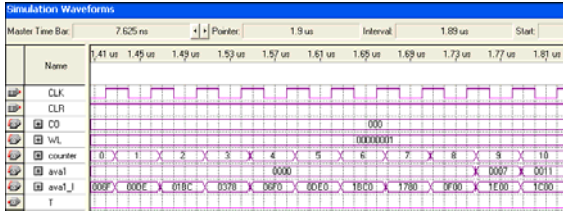
Fig. 5. Eight Vietnamese letters and their PPED vectors (only horizontal and vertical components).

Fig. 6 shows the simulation waveform for the X vector (= T_4) and the T_1 vector:

$X = [00h, 00h, 00h, 03h, 08h, 2dh, 3ch, 3eh, 2ch, 44h, 34h, 1bh, 06h, 00h, 00h, 00h, 00h, 00h, 11h, 1fh, 0ah, 0ch, 08h, 0dh, 10h, 17h, 35h, 0dh, 20h, 53h, 00h, 00h]$.

$T_1 = [00h, 00h, 00h, 03h, 08h, 2dh, 35h, 34h, 2eh, 36h, 30h, 1bh, 06h, 00h, 00h, 00h, 00h, 00h, 00h, 0ah, 0ch, 08h, 0dh, 10h, 17h, 35h, 0dh, 20h, 53h, 00h, 00h]$.

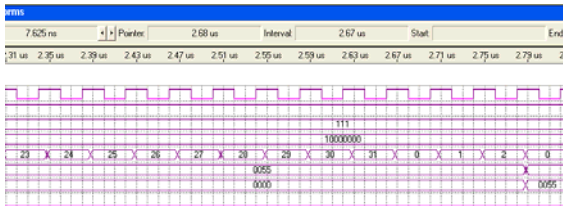
The AVA block gets the Manhattan distance after the 34th clock ($aval = 0055h$), sends to the WTA circuit ($aval_1 = 0055h$) and then is cleared and starts with new data. The output of the WTA circuit is valid after 13 clocks and the T_4 vector is the best match (the output code is 011).



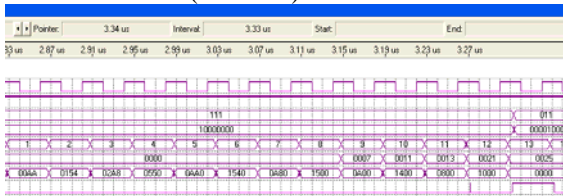
(a). The simulation waveform from clock 0 to clock 10.



(b). The simulation waveform from clock 11 to clock 22.



(c). The simulation waveform from clock 23 to clock 34 (= 31 + 3) and return 0.



(d). The simulation waveform from clock 1 to clock 14.

Fig. 6. The simulation waveform.

After simulation, the result is also tested on the Stratix 672 SmartPack FPGA using EP1S10F672C6ES [5] and data is downloaded by PCI Development Board [6].

4. CONCLUSION

A general VLSI architecture for calculating Manhattan distances has been designed and the number of elements of vectors can be easily changed for different applications. The simulation of the architecture on the Statix for the simple case of Vietnamese letter recognition shows that the architecture works. The architecture is fundamental to develop image recognition and compression

applications by hardware. We will improve the architecture and use a larger FPGA to perform more complex image recognition and compression.

REFERENCES

1. T. Kohonen, self-organization and Associative memory, Berlin, Gemany: Springer –Verlag, 1984.
2. M. Yagi and T. Shibata, An Image representation algorithm compatible with neural-associative – processor-based hardware recognition systems, IEEE transactions on neural networks, vol. 14, September 2003.
3. Khalid Sayood, Introduction to data compression, Morgan Kaufmann Publishers (2000), pp. 257-305.
4. A. Nakada et al., A fully parallel vector quantization processor for real-time motion picture compression, IEEE J. Solid-State Circuits, vol. 34, June 1999.
5. Stratix SmartPack 672 @www.parallax.com
6. PCI Development Kit @www.legacy.memec.com