Algorithm for Image Recognition on FPGA

Rosen Spirov[1]

Abstract **In this paper the author is presented the hardware application of the Haar algorithm for pattern recognition with the FPGA. The project is used the Altera DE2 board to implement a simple hardware design. For describing its behavior use the VHDL language and the Altera's Quartus tools to synthesize and to program the FPGA device**

Keywords **– Image processing, Pattern recognition, Haar wavelet, FPGA, VHDL**

I. INTRODUCTION

In this paper is presented basic hardware architecture, using wavelet Haar coefficients for face recognition.The kernel module is the parallel implementation of processing pixel data. The proposed model is implemented using VHDL and simulated and synthesized into an FPGA. To explain how different parts of the algorithm are tailored must to take advantage of FPGA architecture.

II. **U**sing Wavelet Transforms

The image recognition algorithm consist of several fairly independent stages:

- Wavelet-transform image,
- Filtering in the wavelet domain;
- Fining and Locating the area containing the image;
- Splitting the image into segments and contrasting fragments;
- Recognition of the fragment with pre-trained artificial neural network.

This document is presented basic hardware architecture, using wavelet Haar coefficients for face recognition. Kernel module is the parallel implementation of processing pixel data. The proposed model is implemented using VHDL and simulated and synthesized into an FPGA. To explain how different parts of the algorithm are tailored must to take advantage of FPGA architecture. The application of wavelet - transformation and subsequent filtration allows to recognize this key images with little contrast, such as numbers of vehicles, persons and background, distinguish underwater objects [1]. The problem with recognition of objects in background, that are two-dimensional objects, looks much more complicated than filtering one-dimensional spectra. So, first of all, it is necessary to reduce the dimension of the object, fixing the scale of the wavelet transform in the value determined by the size of the object recognition - the numbers on the plate, incidence of recurrence of black and white in this area and extent of the chosen wavelet. A good result is given symmetric wavelet, which in this case are koyflets [2, 3]. The koyflets is a special version of the fast wavelet-transformations (DWT).

Any function f in $L^2(R)$ can be expanded at some given. Level of resolution j,k in a number of species [3] is given in Eq.1:

$$f = \sum_{k} S_{jn,k}\varphi_{jn,k} + \sum_{j\geq jn,k}^{3} d_{jk}\psi_{jk} \qquad (1)$$

The wavelets $s_{j,k}$ и $d_{j,k}$ can be calculated by the Eq.2 and Eq.3:

$$s_{j,k} = \int f(x)\varphi_{j,k}(x)\cdot dx \qquad (2)$$

$$d_{j,k} = \int f(x)\psi_{j,k}(x)\cdot dx \qquad (3)$$

Once you select a particular wavelet, can be carried out wavelet transform of a signal f (x), defined as orthonormal wavelet basis. Any function $fL^2(R)$ is completely characterized by its wavelet decomposition coefficients on this basis. The function f (x) can be viewed on any n-th level resolution j_n. Then the separation between is average values at this level and fluctuations around them looks like [2, 3] in Eq.4:

$$f(x) = \sum_{k=-\infty}^{\infty} S_{jn,k}\varphi_{jn,k}(x) + \sum_{j=jn}^{\infty}\sum_{k=-\infty}^{\infty} d_{jn,k}\psi_{jn,k}(x) \qquad (4)$$

On an infinite interval, the first sum can be omitted, and the result is a "pure" wavelet decomposition. Coefficients $s_{j,k}$ and $d_{j,k}$ contains information about the composition of the signal at different scales. They can be calculated directly using formula (2), (3). However, this algorithm is not practical, since the calculation will need to spend a lot of (N^2) operations, where N denotes the number of available values of the function. The author describe a much faster algorithm. The Multistage analysis is leads naturally to a hierarchical and fast scheme for calculating the wavelet coefficients of a given function [4]. In general, the iterative formula fast wavelet transform have the form [3,5] is given in Eq.5, Eq.6, Eq.7:

$$s_{j+1,k} = \sum_{m} h_m s_{j,2k+m} \qquad (5)$$

$$d_{j+1,k} = \sum_{m} g_m s_{j,2k+m} \qquad (6)$$

$$s_{j,k} = \int f(x)\varphi(x-k)\cdot dx \qquad (7)$$

[1]Rosen P. Spirov is with the Faculty of Electronics, Technical University-Varna,
1,Studentska str . 9010 Varna , Bulgaria, E-mail: rosexel@abv.bg

These equations provide fast algorithms (the so-called pyramid algorithms) computation of wavelet coefficients, since they require now only $O(N)$ operations for its completion.The simplest solution is the direct use values of $f(k)$ of the available data set in the form of the coefficients $s_{j,k}$ and the application of fast wavelet transform using equations (6), (7). This is a safe procedure, since the pyramidal algorithm provides a complete reconstruction of the signal, and coefficients, in fact, represent the local average values, that was weighted by scaling function [6].

II. REALIZATION

The project includes the creation of parallel models respectively Matlab environment and secondly in Qartus tools. On the figure 1 is shown the Altera DE2 board.

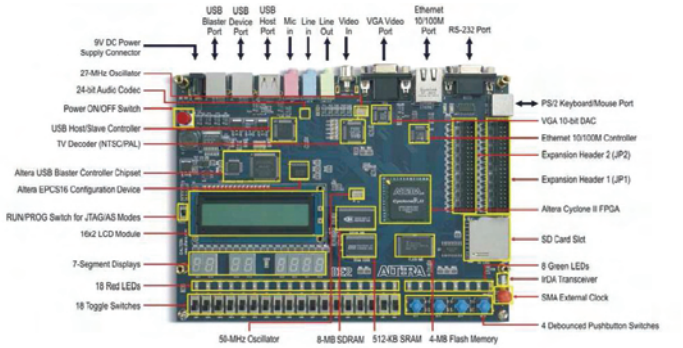

Fig 1. The Altera DE2 board

It consists of an Altera Cyclone II FPGA connected to a variety of peripherals including 512K of SRAM, 4MB of Flash, 8MB of SDRAM, VGA output Ethernet, audio input and output, and USB ports. There are three USB connectors on the top of the board. Parallel architecture of Haar Wavelet (8-inputs) are given in Fig 2:

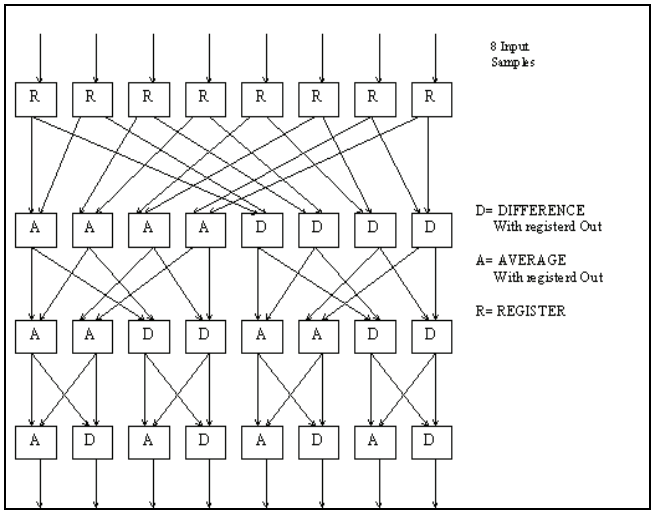

Fig 2. Parallel architecture of Haar Wavelet (8-inputs).

The Simulation Result showing data at various stages at each clock edge is shown in Fig 3.

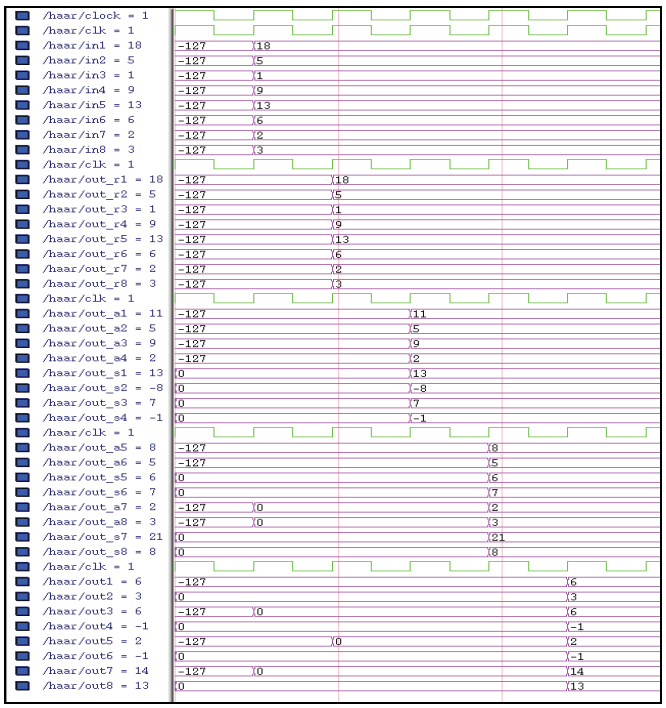

Fig. 3. Simulation Result showing data at various stages at each clock edge

The simulation result showing the parallel nature of the architecture are given in Fig 4:

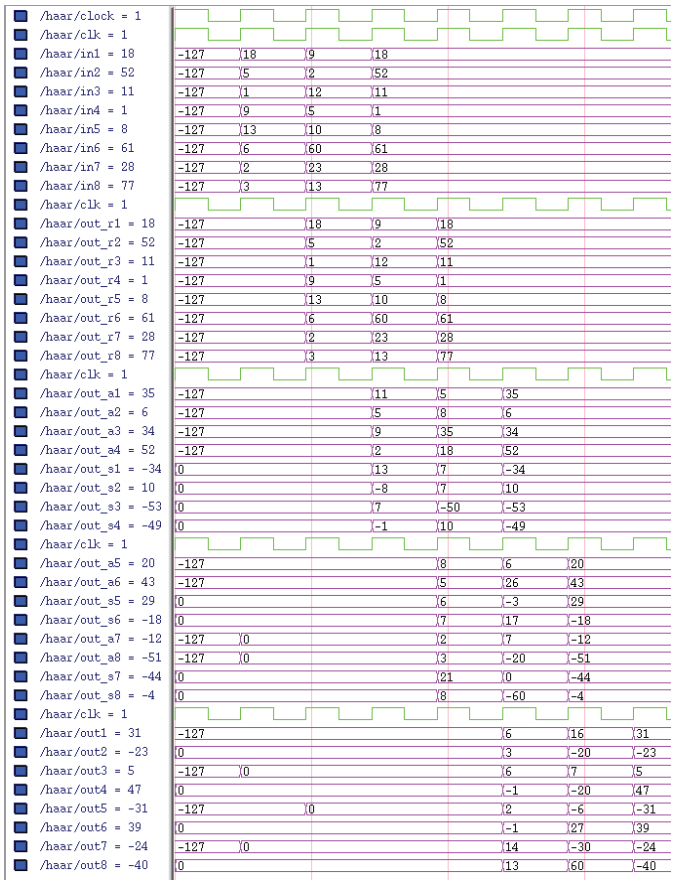

Fig 4. Simulation Result showing the parallel nature of the architecture.

This thesis was focused on design methodology for automatic learning of the optimum transformation of the input space based on Haar wavelet structure. Although the described architecture implements only the Haar wavelet, it can be used as a core subsystem in a preprocessing system which iteratively uses Haar wavelet, selects corresponding coefficients and reapply the Haar wavelet transform. Full realization of the presented concept in FPGA architecture was a subject of another thesis [5]. In abbreviated form is shown in Annex A and Annex B to realize the software Haar wavelet for image recognition and simulation in Matlab of the Quartus VHDL. Source source of Quartus is programming of FPGA on Altera kit. The resources of the FPGA used:

Number of CLBs	120/400 (30%)
Number of bonded IOBs	129/129 (100%)
Number of global buffers	1/12 (8%)
Total equivalent gate count	3948
Minimum period	25.405 ns
Maximum frequency	39.362 MHz
Maximum net delay	10.365 ns
Avrg. Con. Delay	3.494 ns
Avrg. Con. Delay on crt.nets	0.000 ns
Average Clock Skew	0.248 ns
The Max. Pin Delay	10.365 ns
Avrg. Con.Delay on the 10 Worst Nets	9.184 ns

III. SOFTWARE IN AN ABBREVIATED FORM

Annex A. Program in MATLAB: haar.m

```
%plotting the interpretation of haar transform
numCoeff=8;
% determine levels????
levls=round(log10(numCoeff)/log10(2));
x=1:numCoeff;
rows=numCoeff;
coeffMatrix=diag(ones(1,numCoeff));
coeffD=coeffMatrix;
for i=1:levls
coeffC=[];
begn=1;
while begn<numCoeff
oddr=(begn:2:begn+rows-1);
evnr=(begn+1:2:begn+rows-1);
coeffA=[];
coeffB=[];
for j=1:length(oddr)
coeffA(j,:)=( coeffMatrix(oddr(j),:) + coeffMatrix(evnr(j),:)
)/2;
coeffB(j,:)=coeffMatrix(oddr(j),:) - coeffMatrix(evnr(j),:);
end;
coeffC=[coeffC;coeffA;coeffB];
……
```

Annex B. Program in VHDL: Haar.vhd

```
LIBRARY ieee;
USE ieee.ALL;
ENTITY haar IS
PORT (
clock : IN bit;
-- Inputs
in1 : IN integer RANGE -127 TO 127;
……
in8 : IN integer RANGE -127 TO 127;
--Outputs
out1 : OUT integer RANGE -127 TO 127;
……..
out8 : OUT integer RANGE -127 TO 127
);
END haar;
ARCHITECTURE haar OF haar IS
COMPONENT bufgs
PORT (
i : IN bit;
o : OUT bit
);
END COMPONENT;
COMPONENT regPORT (
input : IN integer RANGE -127 TO 127;
clk : IN bit;
output : OUT integer RANGE -127 TO 127
);
END COMPONENT;
COMPONENT adddiv
PORT (
a : IN integer RANGE -127 TO 127;
b : IN integer RANGE -127 TO 127;
clk : IN bit;
c : OUT integer RANGE -127 TO 127
);
END COMPONENT;
COMPONENT difference
PORT (
a : IN integer RANGE -127 TO 127;
b : IN integer RANGE -127 TO 127;
clk : IN bit;
c : OUT integer RANGE -127 TO 127
);
END COMPONENT;
SIGNAL out_r1, out_r2, out_r3, out_r4, out_r5, out_r6,
out_r7,
out_r8 : integer RANGE -127 TO 127;
SIGNAL out_a1, out_a2, out_a3, out_a4, out_a5, out_a6,
out_a7,
out_a8 : integer RANGE -127 TO 127;
SIGNAL out_s1, out_s2, out_s3, out_s4, out_s5, out_s6,
out_s7,
out_s8 : integer RANGE -127 TO 127;
SIGNAL clk : bit;
BEGIN
xbufgs: bufgs
PORT MAP (
clock,
clk
);
-- Input to Register here
r1: reg
```

```
PORT MAP ( in1, clk, out_r1 );
……..
PORT MAP ( out_a7, out_a8, clk, out6 );
a12: adddiv
PORT MAP ( out_s7, out_s8, clk, out7 );
s12: difference
PORT MAP ( out_s7, out_s8, clk, out8 );
END haar;
```

The performance of this implementation can be attributed to the parallel hardware blocks used in performing the necessary calculations for the algorithm [7]. Further to this, the design can be scaled for larger databases by simply adding more processing elements in parallel The above hardware design was implemented on an Altera Quartus II board (clocked at 100 MHz) and was able to perform face recognition on a database of 10 faces in 3.88 milliseconds. A total of 7,820 logic elements were used, 2,348 of which were flip-flops. Again, performance can be attributed to the highly parallel nature of the hardware design and the composite algorithm used FPGA. Original image after wavelet-transform at Haar basis shown on Fig 5. and Fig 6.

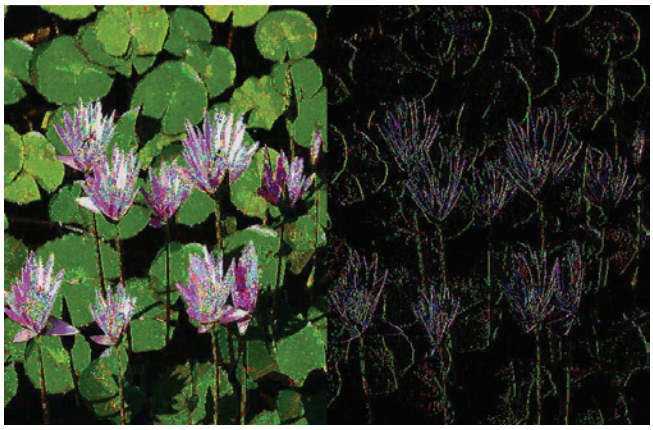

Fig 5. Original image after wavelet-transform by rows at Haar basis.

Fig 6. Original image after wavelet-transform in lines Haar basis

The experiment result shows that the whole operation time is about 60 clock cycle, which about 0.6us at 100MHz clock pulse, so the operation speed can be up to 1.5MHz. The whole design requires 2592 ALUTs and 241 registers (occupancy of resources is about 21%). The advantage of parallel processing in FPGA leads to a substantial increase in performance and accuracy in processing, extraction of the contour information than in the simulation in Matlab.

IV. Conclusion

So it provides a practical method for using FPGA to realize face recognition. The main contribution of our work is design and implementation of a physically feasible hardware system to accelerate the processing speed of the operations required for real time face recognition. The FPGA implementation and simulation results are given in this paper.. The proposed models are implemented using VHDL, and simulated and synthesized into a single FPGA. Therefore, optimization of the hardware source usage was the primary aim of this study. The research results show that the system works with high speed while consuming small amount of logic resources in comparison with original subspace feature extraction process methods. It demonstrates that this technology can produce effective and powerful applications for face recognition systems.

References

[1] P. Belhumeur, J. Hespanha, D. Kriegman. Eigenfaces vs linear projection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1997, *19*(7): 711-720.

[2] H.K. Ekenel, B. Sankur. Multiresolution face recognition.*Image and Vision Computing*, 2005, *23*(5): 469-477. fisherfaces: Recognition using class specific

[3] M. Turk and A. Pentland. Eigenfaces for recognition. *J. of Cognitive Neuroscience*, 1991, *3*(1).

[4] Ahn J.H., Choi S., Oh J.H. A new way of PCA: Integrated-squared-error and EM algorithms. *In: Proc. IEEE Int'l Conf. Acoustics*, *Speech* and *Signal Processing*, Montreal, Canada, 2004.M. S. Sadri and et al. An FPGA based fast face detector *Proc. of GSPX Conference*, 2004: 586-591

[5] .Pellerin D. and S. Thibault. Practical FPGA programming in C. *Prentice Hall PTR*, ISBN: 0-13-154318-0.

[6] Wang Y., Osterman J. and Zhang Y. Video Processing and Communications. C.*Prentice Hall PTR*, ISBN: 0-13-017547-1.

[7] Parhi K. VLSI Digital Signal Processing System Design and Implementation Wiley Inter Since 0471241865.