Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 15 (2011) 2819 – 2823

Procedia Engineering

www.elsevier.com/locate/procedia

Advanced in Control Engineering and Information Science

Performance Driven FPGA Design Analysis with ASIC Perspective of Vector Quantization

Krupa R. Rasane[a], R. Srinivasa Rao Kunte[b]

Department of Electronics and Communication Engineering, KLESCET, Belgaum, 590 008, India
[b]*Principal, JNNCE, Navle, Shivamogga, Karnataka, India*

Abstract

The paper presents and analysis two different approaches for the design of modular and optimized Vector Quantisation (VQ) architectural blocks for the generation of the winner Index from a set of Code vectors giving the best match to the input image vector. The two proposed methods are then analysed for area, logical and routing delays and the ease of redesign to suit specific requirements on the FPGA platform. If an FPGA design need be ported to an ASIC at a later stage it is also important to take this into account early in the design cycle so that the ASIC porting will be efficient. The structured and hierarchically design of VQ lead to the consideration of the dedicated parallel architectures in two ways, in the direction of Codebook size 'N' and the vector dimension 'K', that process with high efficiency. Flexibility in terms of Codebook size expansion and Input vector dimension makes the new design fast to re-configure and meet the specific and challenging needs of a single functioned, tightly constrained and real time VQ encoder. The soft IP core design was targeted on to Xilinx Virtex-5. Simulation result analyses indicate that the design with parallelism only in the direction of the codebook size 'N' gives the desirable performance and is cost effective even for an ASIC implementation performed in Cadence tool. The design satisfies the performance requirements of for a real time image processing. Reconfiguring the codebook size and vector dimension makes redesign simpler to suit applications requirements but at the cost of re-synthesis and is capable of even multiplexed image transmission due to reduction in the Bandwidth.

Selection and/or peer-review under responsibility of [CEIS 2011]

Keywords: FPGA, Vector quantization, Real Time Image compression, ASIC

* Krupa R. Rasane. Tel.: +091-0831-3298710 ; fax: +091-0831-2441644 .
E-mail address: kruparasane@hotmail.com .

doi:10.1016/j.proeng.2011.08.531

1. Introduction

Digital image processing is an ever expanding and dynamic area with applications reaching out into our everyday life such as medicine, space exploration, surveillance, authentication, automated industry inspection and many more areas with different requirements with respect to bandwidth and quality of recovered information. Despite the technological advances in storage and transmission, the demand placed on the storage capacities and on bandwidth of communication exceeds the availability. Vector quantization is a lossy compression technique used to compress picture, audio or video data. A vector quantizer maps *k-dimensional* vectors in the vector space into a finite set of vectors. Each vector is called a Codevector and the set of all the Codevectors is called a *Codebook[1].*

The paper analysis issues like modularity in design, different approaches for easy porting on to an ASIC platform for an optimum Codebook size 'N' or the vector dimension 'K' . It elaborately analysis the architectural issues related to parallelism both in the direction of codebook size 'N' and the vector dimensionality 'K'. The designs in [4][5][6] though modular in nature had larger area and power and hence not suitable for ASIC implementation. The present paper is an optimized version of [5]. The remainder of the paper is organized as follows. Section 2, shows the Metrics used for VQ Analysis, Section3 the hardware realization of the proposed VQ for the two architectures. Section 4 discusses the Result Section followed by conclusion.

2. Vector Quantization Analysis

Two of the error metrics used to compare the various image compression techniques are the Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR). The MSE is the cumulative squared error between the compressed and the original image, whereas PSNR is a measure of the peak error

$$MSE = 1/MN \sum_{Y=1}^{M}[I(X,Y)-I'(X,Y)]^2 \qquad (1)$$

$$PSNR = 20\ Log\ 10(255/SQRT(MSE)) \qquad (2)$$

where I(X,Y) is the original image, I'(X,Y) is the approximated version (which is actually the decompressed image) and M,N are the dimensions of the images. Logically, a higher value of PSNR is good because it means that the ratio of signal to noise is higher. Here, the 'signal' is the original image, and the 'noise' is the error in reconstruction.

3. Hardware Realization of 4 Codevector Vector Quantization

An analysis made from the MATLAB [4] shows that more than 50% time is spent on processing the MSE as in equation 1. In this paper, VQ encoder can be suitably designed with parallelism and pipelining on to an FPGA. The Image can be stored on or off the chip memory. Also, two different architectures are considered for porting on to an ASIC platform with optimum performance.

3.1. Image Preprocessing Module

An Image of size M x M is divided into 'm x n' pixels called as vector of 'K' dimension represented as

$$X_m=(X_{0,}X_1,X_{2,}X_3\cdots\cdots\cdots X_k) \qquad (3)$$

A trained Codebook of similar Dimension and size 'N' is obtained by training a number of standard images. Let the m^th Codevector be represented as follows

$$C_m = (C_{m0,} C_{m1}, C_{m2}, C_{m3} \cdots\cdots\cdots C_{mk}) \qquad (4)$$

Let each Codevector be associated w.r.t an Index *IN* i.e. for a size of 4, I0 is associated with CW0, I1 with Codevector CW1, I2 with Codevector2, and I3 with Codevector3.

3.2. Proposed Hierarchical Design Approach and Optimization

The proposed model is built hierarchically using a basic VQ1 component, which is implementing for a 'K' Dimensional Codevector of Codebook size 'N = 4. VQ of a suitable Codebook size 'N' is done by parallel reuse of VQ1 that forms the first stage of the system. VQ is processed one vector at a time, generating the winner Index for all 'm x n' vectors. The pipelining ensures that the winner index is generated every clock cycles for each input vector to achieve the required throughput of 30 frames/sec in real time applications. Optimizing an algorithm for an ASIC is common to many of the optimized design for FPGA except ASIC will be implemented at gate level and FPGA using the resources on FPGA. The proposed architecture is analysed for two approaches in this paper as follows, refer Fig 1. (a)

(i) Parallel only in the direction of 'N' and sequential with pipelining in the direction of 'K' for VQ.
(ii) Parallelism in the direction of 'N' and vector dimension 'K' with pipelining.

3.3. Implementation Details

Our design Implements the MSQ for a 4 Dimensional 4 CWD with prior computation of known values and for an input of 8 bits per pixel and a Codebook of size N=256**.**

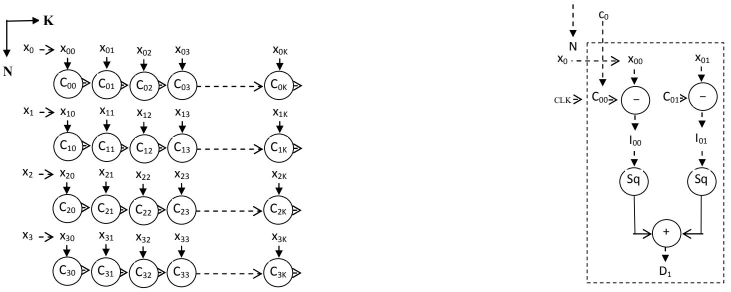

Fig. 1. (a) Parallelism in the direction of Codebook size 'N' and vector Dimension 'K'; (b) 2D distortion pipelining

3.4. Parallel only in the direction of 'N' processed sequential with pipelining in the direction of 'K'

For a VQ of 'K' Dimensional Input vector, the MSE Equations is modified to introduce the pipelining for the optimizing throughput, since most FPGA designs are not latch/FF limited and the distortion error for every input and Codevectors is pipelined with parallelism in direction of 'N'. The errors are then squared as seen in Fig 1. (b). Distortion D1, D2, D3, D4 is evaluated as seen in Fig 2. These are compared individually to generate Boolean values for A1, A2, and A3. A look-up-table [5] is then used to determine the Winner Index 'Ix' and the least distortion 'Dx' among the 4 Codewords of VQ1. The proposed design focuses on decreasing logic usage. All tools in the FPGA design flow have many options that will impact the maximum frequency, resource utilization, power usage, and sometimes even the correctness of the final design.

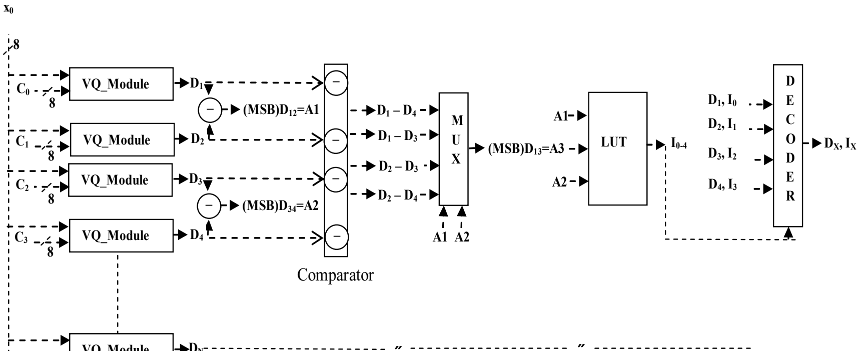

Fig.2. Hardware Process flow for VQ on FPGA

Finding the optimal choices for a certain design was not an easy task. It is also not uncommon that the logical choice is not the best solution. Lastly, both ASIC and FPGA have their own advantages and disadvantages and the suitability of the said platform depends upon the volume of production, performance, cost and time to market. An FPGA, is superior for rapid prototyping, low set up cost, configurability etc. while an ASIC is performance efficient.

3.5. Parallel in the direction of 'N' and vector dimension 'K' with pipelining

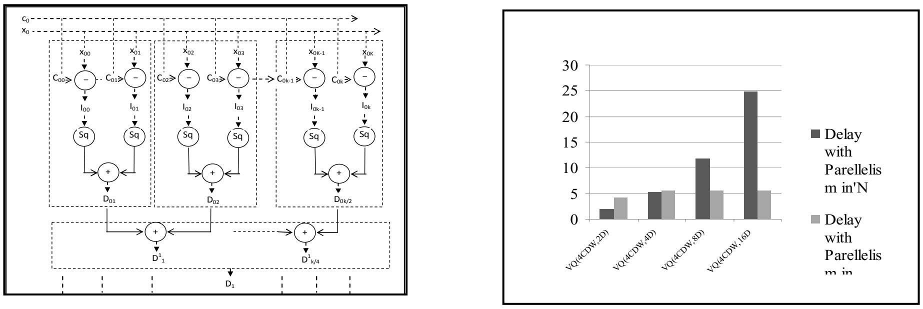

Fig. 3. (a) VQ with Parallelism both in the direction of 'N' and 'K'; (b) Delay Comparison shown for the two architecture

The proposed model observes parallelism in the direction of both 'N' and 'K' as depicted in Fig 3. (a) This approach evaluates the MSE hierarchically by evaluating the individual MSE then added in the next stage to evaluate the MSE between the individual input vector and the Codevector.

4. Result

The VQ was tested for different values of 'N' and 'K' for both the two approaches, one using the parallelism in the direction of 'N' and the other in the direction of both 'N' and 'K'. The results show that both approach have approximately constant delay for a given vector dimension 'K' for variable codebook size 'N', however routing delay increases as compared to logic delay as seen in Fig 4 (a). Also, as vector dimensionality increases the processing time for a 4CWD increases, for parallelism in 'N' whereas it is constant for parallelism in both the direction of N and K, but at the cost of design complexity and more resources as depicted in Fig 3. (b).

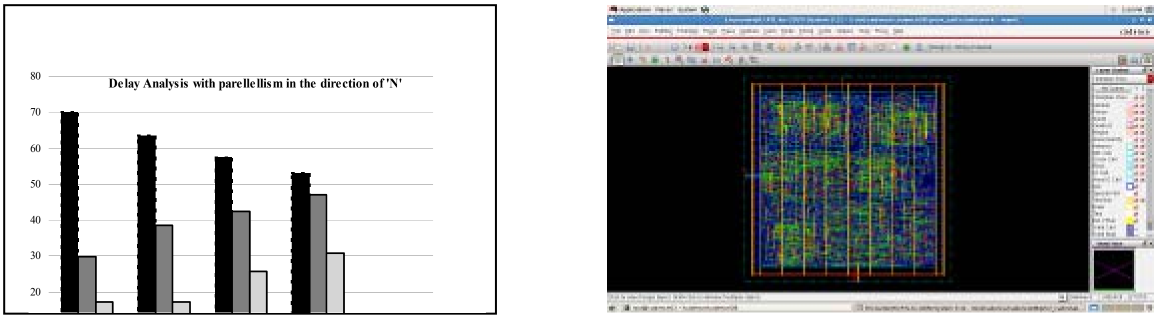

Fig. 4. (a) Chart showing the delays associated with delay types with different values of ‘N’; (b) ASIC Floor plan

5. Conclusion

The paper shows two design issues analysed in order to decide as to which approach is better suitable for an optimized VQ ASIC implementation. The results discussed shows that the parallelism in the direction of N is an optimum solution among the two for an ASIC design, as chip area reduces, because of reduced design complexity and resources required. Such a VQ chip was designed for N=16, k=2, using the Cadence tool which satisfied the real time requirement of 30 frames/sec. The chip has cell instances of 6654 and cell area 49441, 3356752.872mW power as compared to 3.8W on FPGA, see Fig 4. (b).

Acknowledgements

The Principal and Head of the Electronics and Communication Department, KLESCET, Belgaum. Head of Computer Science Department, JNNCE, Shimogga for their suggestions and VTU for providing laboratory facilities which has helped us to bring out this paper in time for submission.

References

[1] Ali M. Al-Haj, An FPGA-Based Parallel Distributed Arithmetic Implementation of the 1-D Discrete Wavelet Transform, Informatics 29 (2005) 241–247, February’07, 2004.

[2] P. Y. Chen, R. D. Chen, An index coding algorithm for image vector quantization. IEEE Transactions on Consumer Electronics, vol. 49, no. 4, pp. 1513-1520, Nov. 2003.

[3] Krupa Rasane, Srinivasa Rao Kunte, Speed Optimized LUT based ‘K’ Dimensional Reusable VQ Encoder Core, IEEE Explore, and pp: 173 - 178 Digital Object Identifier: 10.1109/ICECTECH.2010.5479963.

[4] Krupa Rasane, Srinivasa Rao Kunte, Design and Implementation Issues of Parallel and Pipelined Vector Quantization in FPGA for Real Time Image Processing, Communications in Computer and Information Science (ISSN:1865-0929, Book Series) included in Springer Digital Link Library, Volumn 135, 2011, DOI:10.1007/978-3-642-18440-4, Part 3, Pgs 366-371.