Available online at www.elixirpublishers.com (Elixir International Journal)

Electrical Engineering

Elixir Elec. Engg. 61 (2013) 17123-17127

ISSN: 2229-712X

Hardware implementation of high speed low complexity variable block search motion estimation for video compression

D.V. Manjunatha[1,*] and G. Sainarayanan[2]

[1]Department of EEE, New Horizon College of Engineering, Bangalore.
[2]Visveswaraya Technological University, Belgaum, Karnataka, India.

ARTICLE INFO

Article history:
Received: 18 June 2013;
Received in revised form:
24 July 2013;
Accepted: 8 August 2013;

Keywords
VC, ME,
VBSME, PD,
FPGA, HDTV,
PE etc.

ABSTRACT

The key factor in video processing is the Video Compression (VC) which makes video signal to be processed and stored efficiently. Video compression involves complex algorithms such as Motion Estimation (ME). The hardware implementation of motion estimation has a trade-off between area, speed and Power Dissipation (PD). This paper implements high speed low complexity Variable Block Size Motion Estimation (VBSME) for H.264 standard using verilog coding on Altera Stratix – IV Field Programmable Gate Array (FPGA) (EP4SGX70DF29C2X) device, using parallel and pipeline timing approach to achieve high speed and the low complexity for High Definition Television (HDTV) system. The proposed work computes all 41 vectors of various block sizes during 180 clock cycles with less complexity using only 46 Processing unit Elements (PEs) and achieves a maximum operating frequency of 486.62 MHz.

Introduction

The communication industry is growing enormously in the field of multimedia due to its advances in digital communication over a few decades. Many application areas have emerged, because of this important characteristic. For example, video-on-demand, video-phone, set-top box, video playback using compact disk, video conferencing, point to point video delivery and mobile TV broadcasting, etc. The development of video processing systems having different size, quality, power consumption and cost were led by the specialized nature of video applications, such applications require video compression with ever higher compression ratios, better visual quality and high bandwidth.

In general the development of hardware architectures are designed to form the integrated circuits which allows parallel processing of data from various sub blocks of the architectures, however hardware architectures suffers from limitations such as algorithm flexibility due to timing dependencies, which arises from data-flow between various blocks of the architecture. Thus the development of good architecture for video codec in integrated circuits is very important, as the customization of video codec is nothing but video compression in the modern state of the art of Digital VLSI systems. Thus the realization of efficient video compression block is very important to achieve high quality, performance, storage and transmission capabilities.

The compression ratio plays an important role in the field of video processing. The motion in the video scene will reduce the efficiency of the compression ratio. The efficiency of the compression ratio can be increased by exploiting the similarities between the video frames. This exploitation of similarities can be done using the motion estimation, hence the motion estimation field has seen the highest research topic and interested issue in the past a few decades. In short the motion estimation means the estimation of the displacement (or velocity) of image structures from one frame to another in a time sequence of 2-D images of the video, in order to achieve video compression in video coding.

Video codec is the customized method for achieving the video compression, the function of video codec is to do compression and decompression of video sequence of data for achieving good storage and transmission. In the recent days general purpose circuits have improved in terms of performance, reliability, low cost and are increasingly available hence realization of software video codec for applications in multimedia / video conferencing etc. has become very easy, but recently available software codec suffers from limited compression performance, speed and quality functionality due to limitations in the processing resources.

Video codec implemented on the general purpose processor such as personal computers can compete with the processes for the resources. In recent trends there are varieties of hand held devices namely cell phones are becoming more versatile by incorporating multiple functions such as telephony, messaging, entertainment, games, mobile computing, photography, internet on mobile and many more applications, so the video codec running on these highly sophisticated hand held devices must be strong enough to compete with other processes for the time, speed and area, in addition such hand held devices must have the good battery to take care of the system, so the constraints here are the low power consumption and the high speed, then only such hand held devices can sustain in the market. Since 90's digital video coding has gradually increased its importance. Video delivery, storage and presentation have large impact on these devices.

Digital video coding achieves higher data compression rates without much loss of picture quality, compared to analog video. The need for high bandwidth as required in analog video delivery is eliminated by this. Digitization of video scenes was

Tele:
E-mail addresses: dvmanjunatha@gmail.com

an inevitable step since it has many advantages over analog video. Digital video is virtually immune to noise, easier to transmit and is able to provide a more interactive interface to users, since the bandwidth required for analog delivery can be used for more channels in a digital video delivery system, furthermore, the amount of video content, e.g. TV content, can be made larger through improved video compression. End users can also stream, edit and share video with friends via internet or IP networks, with today's sophisticated video compression systems. In contrast, manipulation and transmission of analog signals are difficult.

Generally speaking, a technology for transforming video signals that aims to retain original quality under a number of constraints is video compression, e.g. storage, time delay or computation power constraints etc. to reduce the storage requirement by applying computational resources; it takes advantage of data redundancy between successive frames. Normally trade-off between quality, speed, resource utilization and power consumption are involved in the design of data compression systems. In a video scene, from spatial, temporal and statistical correlation between frames the data redundancy arises, because of these differences in their characteristics, the correlations are processed separately. To reduce temporal redundancy between successive frames in the time domain, motion estimation and compensation are used. To reduce spatial dependency within the frame in the spatial domain, transform coding is commonly used. Lastly, to reduce statistical redundancy over the residue and compression data, entropy coding is used. So in this work the high speed, low complexity variable block search motion estimation algorithm is addressed using verilog coding with its hardware (FPGA) implementation.

The rest of the paper is organized as follows: Section II describes the related work in the VBSME to find the motion vectors. The VBSME architecture is described in section III. Section IV describes the VLSI Implementation of VBSME. The results and discussion are presented in section V. Finally we conclude in section VI.

Related Work

In H.264/AVC [1, 2] the coding efficiency is improved approximately about 60% as compared with the Motion Picture Experts Group (MPEG)-2 and H.263 standards [3, 4] since there are advanced motion compensation techniques such as variable block size motion estimation, multi frame reference etc., to improve the coding efficiency. Some fast algorithms of variable block size motion estimation [5–9] are describing the methodologies to reduce the computational complexity and most of these methodologies of fast motion estimation algorithms are applicable only for software implementation, but a some of them can be realizable in hardware implementation, due to their searching routines are irregular and uses unpredictable flow.

The Very Large Scale Integration (VLSI) implementation for VBSME has been proposed in recent years [10–15]. The full search algorithm is still used because of high regularity. However, the circuit complexity is high, in these methods so we need to develop the coding in such a way that our design must give very high speed and it should have less complexity.

In this paper, in order to reduce the chip complexity and keep high regularity, and achieve high speed the full search algorithm is implemented using the parallel structure and pipeline scheduling approach [17] with hardware-oriented approach to estimate variable block modes using verilog [16] coding.

VBSME Architecture

The variable block size motion estimation of H.264/AVC video coding standard each picture frame is divided into many macro blocks. Each macro block is further divided into seven different sub-block sizes they are 4×4, 4×8, 8×4, 8×8, 8×16, 16×8 and 16×16. This VBSME approach can provide better coding performance. However the computational cost is very high as there are 41 vectors to be estimated for processing one macro block. The VBSME structure of macro block is as shown below.

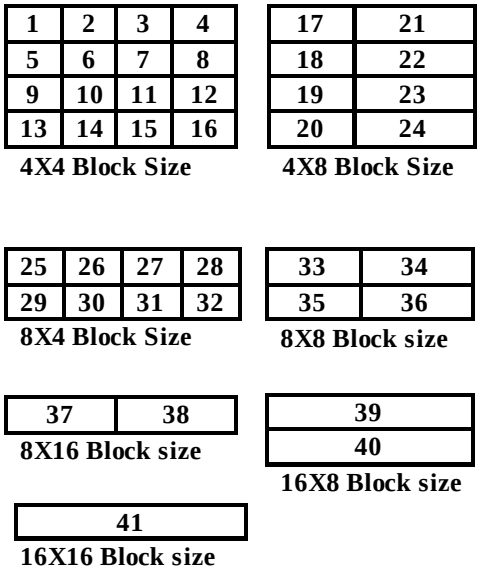

Figure 1: Macro block sizes in VBSME in H.264

Here bottom up approach is used to reduce the complexity of VBSME using the parallel and pipeline approach. The sum of absolute difference of NXN sized block can be computed using four sub blocks with a block size of N/2X N/2 and named as W, X, Y and Z as shown in equation 1.

$$SAD_{NXN} = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|ft(q+i,p+j)-\left(ft-1(q+i+mx,\quad p+j+my)\right)\right|$$

$$= W+X+Y+Z \quad \text{----------1}$$

$$W=SAD_{N/2\ X\ N/2} = \sum_{i=\frac{n}{2}}^{N}\sum_{j=0}^{\frac{N}{2}-1}\left|ft(q+i,p+j)-\left(ft-1(q+i+mx,p+j+my)\right)\right|$$

----------2

$$X= SAD_{N/2\ X\ N/2} = \sum_{i=\frac{N}{2}}^{N}\sum_{j=0}^{\frac{N}{2}-1}\left|ft(q+i,p+j)-\left(ft-1(q+i+mx,p+j+my)\right)\right|$$

----------3

$$Y=SAD_{N/2\ X\ N/2} = \sum_{i=0}^{\frac{N}{2}-1}\sum_{j=\frac{N}{2}}^{N}\left|ft(q+i,p+j)-\left(ft-1(q+i+mx,p+j+my)\right)\right|$$

----------4

$$Z=SAD_{N/2\ X\ N/2} =$$

$$\sum_{i=\frac{N}{2}}^{N}\sum_{j=\frac{N}{2}}^{N}[ft(q+i,p+j)-(ft-1(q+i+mx,p+j+my))|\square] \quad \text{----------5}$$

Where ft and ft-1 are two successive frames, the position (q, p) is the location of current coding MB, the block size is defined using N and mx and my are searching horizontal and vertical vectors respectively the SAD of Four N/2XN/2 sub-blocks can be calculated by equations 2, 3, 4 and 5 as shown above for W, X, Y and Z respectively.

Based on the above concept we can compute SAD of small block and then expand for the larger searching block modes of H.264/AVC. Here first we developed a basic sub-block of 4X4 unit to develop other larger macro blocks of variable block sizes. The figure below shows 16X16 blocks are splitted into 16, 4X4 sub- blocks.

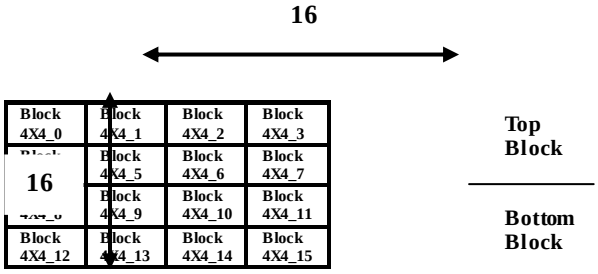

Figure 2: 16X16 Macro block partitioned into 4X4 blocks

When the various vectors of 4X4 blocks are computed then the minimized SAD value is computed. Using 4X4 blocks other motion vectors can also be obtained and the corresponding SAD values of all macro blocks of VBSME can be computed.

Finally the best block mode can be selected corresponding to the minimum SAD value of various motion vectors of VBSME.

VLSI Implementation of VBSME

The VLSI Implementation of VBSME of H.264 / AVC standard uses the basic 4X4 block. The input pixel values are obtained from temporary buffers of 32 bits size as shown in the figure below.

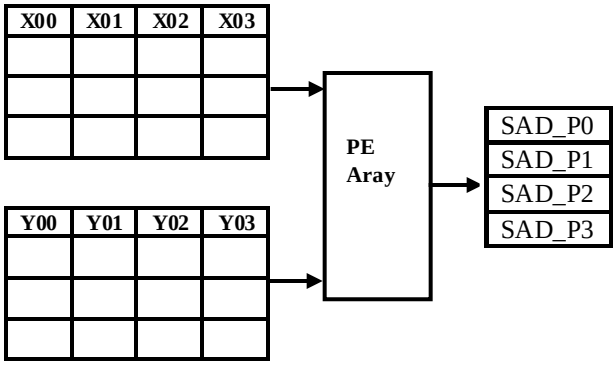

Figure 3: Accessing pixel values from buffers into the processing Array.

In this system four pixels can be read in row by row fashion in one clock cycle. The partial value of SAD is accumulated in the accumulator. After 4 clock cycles one can achieve the complete SAD value for the 4X4 block.

The building block involved in computing the partial SAD value called Processing unit Element (PE), which involves 4 subtractions and 3 additions as shown in figure 4.

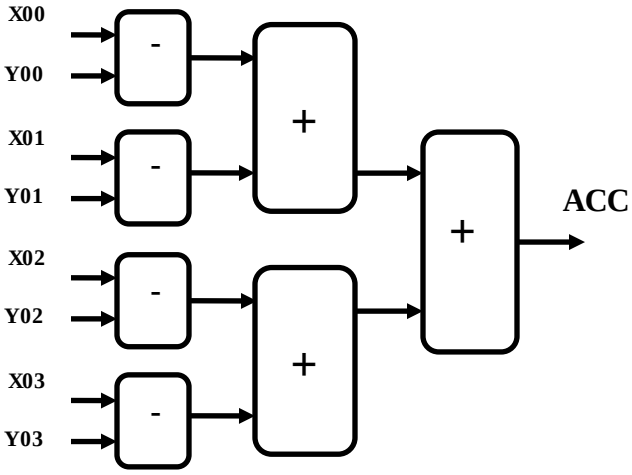

Figure 4: Basic Processing Unit Element (PE)

The current pixels X00, X01….. and the reference pixels Y00, Y01…….. are read in parallel.

Using 4 subtractions and 3 additions we can get the partial SAD from one row per cycle so that the 4 partial SADs needed together gives the entire SAD for one motion vector of 4X4 block. Once the notion vector for 4X4 block is found then using 4X4 block we can compute motion vectors of all the macro blocks of VBSME. The synthesized circuit RTL of processing unit element is as shown in figure 5.

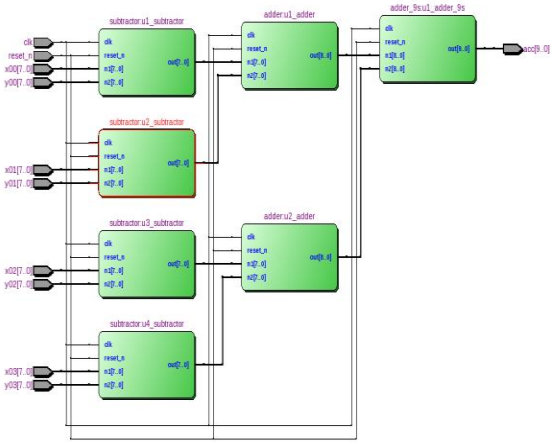

Figure 5: synthesized Processing Unit Element (PE)

Further the Large Processing unit Element (LPE) is used to find the SAD values parallel, one LPE consists of 3 processing units to compute the SAD values parallel. The synthesized LPE architecture is as show in figure 6.

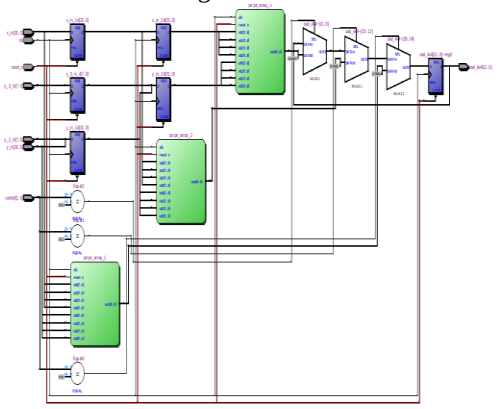

Figure 6: synthesized Large Processing Unit Element (PE) to compute the SAD values in parallel.

Using Large Processing unit Element (LPE) the VBSME synthesized RTL architecture is as shown figure 7.

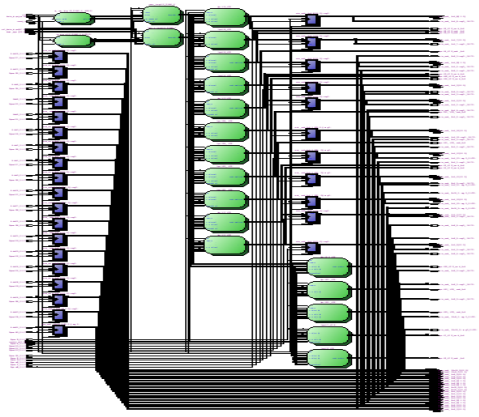

Figure 7: Synthesized VBSME Architecture using Large Processing Unit Element (LPE)

Results And Discussion

Based on the VBSME Architecture proposed in sections III and IV the VBSME chip is implemented using verilog HDL. First each module is simulated using modelsim simulation tool and then each one of the module is synthesized using Altera Quartus II synthesis tool on the Stratix IV FPGA (EP4SGX70DF29C2X) device. In this implementation there are 41 minimum SAD values of different block sizes corresponding to the relative motion vectors. The proposed work computes all 41 vectors of various block sizes during 180 clock cycles with less complexity using only 46 Processing unit Elements (PE) and achieves a maximum operating frequency of 486.62 MHz.

The various simulation results of VBSME and the VBSME Chip are as shown in figures 8, 9 and 10.

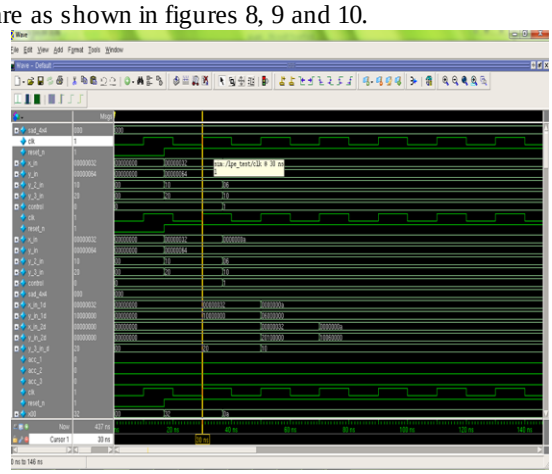

Figure 8: Simulation results of 4X4 VBSME

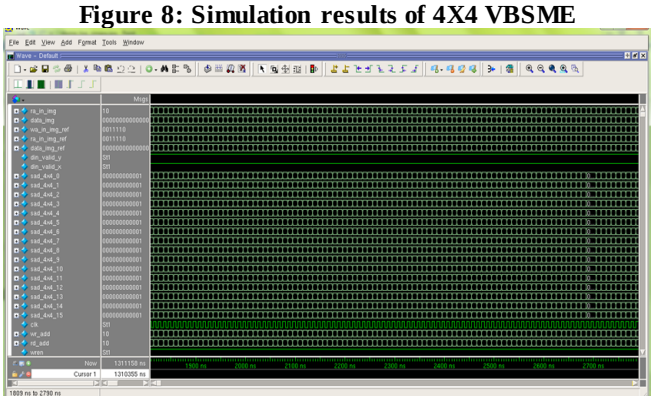

Figure 9: Simulation results of 4X4 partial SAD to get 16X16 SAD- VBSME

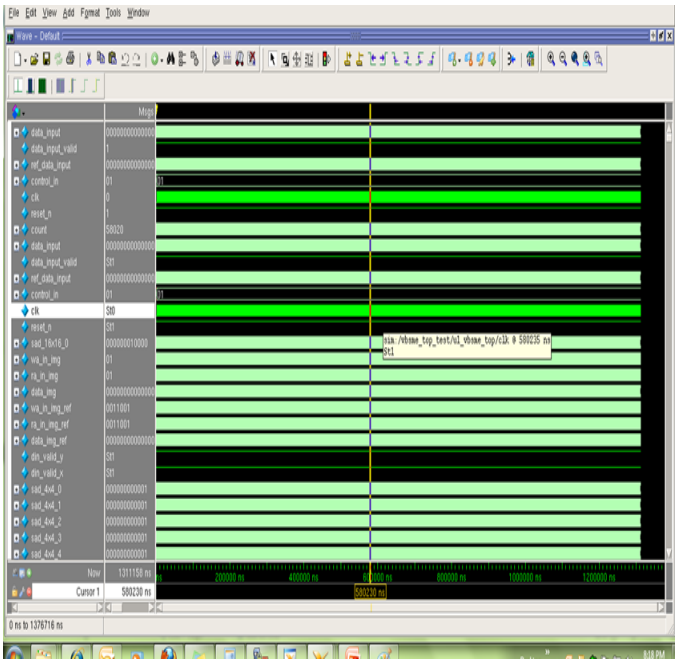

Figure 10: Simulation results of overall VBSME

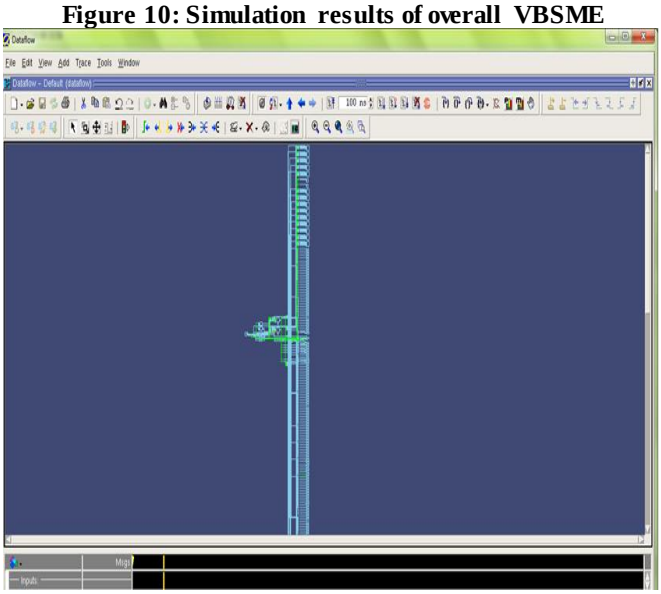

Figure 11: Synthesis results of overall VBSME chip

There are various papers presented in the literature about VBSME architectures. We considered 3 papers for comparison of results with the proposed work as shown in table 1 below.

Table 1: Comparison of Proposed work with other architectures

	Ou et al.	Kuo and Chan	Wei et al.	Proposed work
Algorithm	Full search	Full search	Full search	Full search
No. of PEs	256	256	256	46
Search range	16X16	48X32	33X33	72X72
Max Freq. in MHz	200	81	200	486.62
Specs	HDTV	HDTV	SDTV	HDTV

For HDTV coding system, the work proposed by Ou et al [11], Kuo and Chan [12] and Wei et al.[13] requires 256 Processing unit elements(PEs), but the proposed work in this paper outperforms in terms of both speed (462.62MHz) and the complexity (uses only 46 PEs)

Conclusion

This paper presents the FPGA implementation of high speed low complexity Variable Block Size Motion Estimation for video coding using Altera Quartus II synthesis tool, the simulations are done using modelsim simulation tools. The proposed architecture outperforms all the other architectures which compared in the results and discussion section in terms of speed and the complexity. The presented work computes all 41 vectors of various block sizes during 180 clock cycles with less complexity using only 46 Processing unit Elements (PEs) and achieves a maximum operating frequency (speed) of 486.62 MHz. for HDTV applications of H.264 / AVC video coding standard.

Acknowledgment

The authors would like to thank Prof Venkatesvarlu and Prof. Siva Yellampalli of UTL technologies for their support in the lab.

References

[1] H.264/AVC, "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-TRec.H.264/ISO/IEC14496-10AVC", in 'Joint Video Team (JVT) of ISO/IECMPE Gland ITU-TVCEG', JVT G050, 2003.

[2] Richardson.I.E.G, "H.264 and MPEG-4 video compression video coding for next-generation multimedia" (John Wiley & Sons, 2003). ISO/IEC DIS 13818-2, MPEG-2 video coder.

[3] ISO/IEC DIS 13818-2, MPEG-2 video coder

[4] Cote G, Erol B, Kossentini F, H.263+, "Video coding at low bit-rate", IEEE Trans. Circuits Systems, Video Technology, 1998, 8, (7), pp. 849 – 866.

[5] Liang L, Mccanny J.V. Sezer S, "Multi-standard sub-pixel interpolation architecture for video motion estimation" IEEE International, SOC Conference, September 2008, pp. 229 – 232.

[6] Lee J, Jeon B, "Fast mode decision for H.264 with variable motion block sizes" IEEE Int. Symposium On Computer and Information Sciences (ISCIS), November 2003, pp. 723–730.

[7] Al Qaralleh E.A, Chang T.S, "Fast variable block size motion estimation by adaptive early termination", IEEE Transactions on Circuits Systems, Video Technology, 2005, 15, (6), pp. 784–788.

[8] Wu D., Pan F., Lim K.P., ET AL.: 'Fast intermode decision in H.264/AVC video coding', IEEE Trans. Circuits Syst. Video Technol., 2005, 15, (6), pp. 953 – 958.

[9] Jing X., Chau L.P, "Fast approach for H.264 inter mode decision", Electronics Letter 2004, 40, (17), pp. 1050 – 1052.

[10] Yap S.Y, Mccanny J.V, "A VLSI architecture for variable block size video motion estimation", IEEE Transactions of. Circuits Systems – II, 2004, 51, (7), pp. 384 – 389.

[11] Ou C.M, Le C.F, Hwang W.J, "An efficient VLSI architecture for H.264 variable block size motion estimation", IEEE Transactions on Consumer Electronics, 2005, 51, (4), pp. 1291 – 1299.

[12] Kuo T.Y, Chan C.H, "Fast variable block size motion estimation for H.264 using likelihood and correlation of motion field", IEEE Transactions on Circuits Systems, Video Technology, 2006, 16, (10), pp. 1185 – 1195.

[13] Weic, Hui H, Jiarong T, Jinmei L, Hao M 'A high-performance reconfigurable VLSI architecture for VBSME in H.264', IEEE Trans. Consumer Electronics, 2008, 54, (8), pp. 1338 – 1345.

[14] Liang L, Mccanny J.V, Sezer S, "Systolic array based architecture for variable block-size motion estimation", Second NASA/ESA Conference on Adaptive Hardware and Systems, 2007, pp. 160 – 168.

[15]Chen W.N, Hang H.M, "H.264/AVC motion estimation implementation on Compute Unified Device Architecture (CUDA)" IEEE International Conference on Multimedia & Expo, 2008, pp. 697 – 700.

[16] PALNITKAR S, "Verilog HDL" (Prentice-Hall, NJ, 1996).

[17] Seetharaman Ramachandran "Digital VLSI Systems Design – A Design Manual for Implementation Projects on FPGAs and ASICs using Verilog" (Springer, 2007).