I J C T A, 9(34) 2016, pp. 673-680
© International Science Press

A High Performance VLSI Architecture with Half Pel and Quarter Pel Interpolation for A Single Frame

K. Priyadarshini[1] and D. Jackuline Moni[2]

ABSTRACT

A high performance VLSI hardware architecture for half-pixel and quarter pixel interpolation which can be implemented for H.264 / MPEG4 Part 10 video coding is designed . The hardware design comprises of two engines, first engine performs half pel interpolation followed by quarter pel interpolation for single video frame. To improve the resolution of the processed frame, second engine performs the interpolation among the location of the quarter pixels pointed out by the engine1.This designed hardware is to be used as part of a complete H.264 video coding system for portable applications. The hardware architecture designed is fully explained in VHDL and implemented in Verilog HDL. The experimental analysis produced good results when compared with previous related works. The Verilog RTL code is verified to work at 200 MHz in a Xilinx SpartranVI FPGA.

Keywords: Motion estimation, Interpolation, Half-pel, Quarter-pel

1. INTRODUCTION

Multimedia communication with efficient standards for video coding are required for the compression of the video content .The reason for this concept is because of the requirement of large number of bits for the transmission of uncompressed video data. The standards assigned for video coding follow a single frame work in terms of the algorithm, but the differences lies in the range of parameters and coding modes. Video coding plays a very important role in the area of research due to the increasing demand for various applications like video storage, digital television broadcasting. Video coding mainly involves to achieve compression to eliminate redundancy in the video data. The two types of redundancy which are present in video data are spatial and temporal redundancy. Removal of temporal redundancy involves looking between frames and called as “Intercoding” while spatial redundancy is removed using various transform coding techniques .Motion compensation involves the removal of temporal redundancy in video sequences. The main idea of motion compensation is to estimate the motion of objects and with this information to build a prediction for successive frames. Commercial products mainly use the concept of video compression for many real time applications. Figure 1 shows the basic block diagram of H.264 Encoder. In this block Motion estimation section is the most important and challenging section. Various international standards have been developed ,considering the application of video compression. In the video encoder section the aspect of motion estimation is the challenging task. To improve the performance of integer pel motion estimation, half-pel variable block size motion estimation followed by quarter pel is carried out.

[1] Research Scholar, Department of ECE, Karunya University, Coimbatore.

[1] Electronics & Communication Engineering, School of Electrical Sciences, Karunya University Coimbatore, 641114, Tamil Nadu, *E-mail: priyanilesh@rediffmail.com*

[2] Professor, Department of ECE, Karunya University, Coimbatore.

[2] Electronics & Communication Engineering, School of Electrical Sciences, Karunya University Coimbatore, 641114, Tamil Nadu, India, *E-mail: moni@karunya.edu*

This work proposes an efficient high performance VLSI architecture for half pixel and quarter pixel interpolation architecture. The designed steps summarized in this paper is fully based on H.264/AVC standard,but focussed towards the simplification and optimization.

The paper flow is structured as follows. Section 2 explains motion estimation process.Section 3 explains the half-pixel interpolation,quarter pixel interpolation search process.Section 4 explains software analysis. Section 5 describes the proposed architecture. Section 6 shows the synthesis result.Section 7 presents the comparison among related works and Section 8 concludes the paper.

2. CONCEPT OF MOTION ESTIMATION

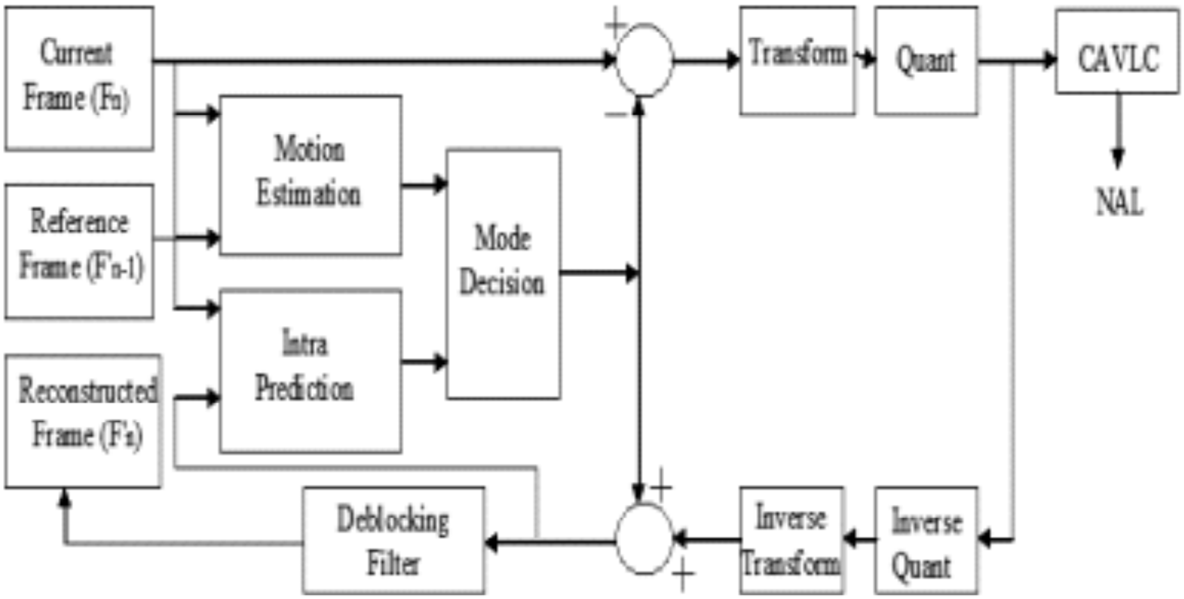


Figure 1: Block Diagram of H.264 Encoder

3. HALF-PEL INTERPOLATION

Half pel means that the pixels are interpolated and new pixels are generated for specific purpose.To increase the motion vector accuracy,quarter pel resolution is used with the concept of interpolation.But in H.264 a 6 Tap FIR filter is used to determine the half-pel resolution,but for quarter pel resolution just normal average is used. H.264/AVC uses the concept of coefficient interpolation at half pel and quarter pel accuracy.In the design 6 Tap wiener interpolation filter is used [1]. The interpolation deals only with horizontal and vertical directions and hence it is not suitable for the textural sequences.To improve coding efficiency in the video coding standard 2D-non-seperable 6-Tap adaptive interpolation filter (AIF) method is used [2].Seperable adaptive interpolation scheme is proposed in [3] to simplify the implementation of non-seperable adaptive interpolation scheme.The optional half pel and quarter pel interpolation is performed with accuracy.Interpolation paves the path for the improvement of the resolution of the image.But in the hardware architecture design ,the interpolation phase is costly in terms of hardware resources [4, 5, 6].

4. QUARTER –PEL INTERPOLATION

Quarter-pixel motion *also known as* ***Q-pel motion*** *or* ***Qpel motion****)* refers to using a quarter of the distance between pixes or luma sample positions as the motion vector precision for motion estimation and motion compensation in video compression schemes. It is used in many modern video coding formats such as MPEG-4,H.264/AVC and HEVC. Though higher precision motion vectors take more bits to encode, they can sometimes result in more efficient compression overall, by increasing the quality of the prediction signal. Quarter-pixel motion compensation much like half-pixel, is achieved through interpolation. Different specific schemes are used in different designs. H.264/AVC uses a 6-tap filter for half-pixel interpolation and then simple linear interpolation to achieve quarter-pixel precision from the half-pixel data. The new features such as variable block-size, quarter sample-accuracy and multiple reference frames increase the complexity and computation load of motion estimation greatly in H.264/AVC encoder. Experimental results have shown that motion estimation can consume 60% for 1 reference frame to 80% for 5 reference frames of the total encoding time of H.264 codec [7]. So far, there have been a very few VLSI implementations [8, 9] for H.264/AVC motion estimation considering variable block size. But none of them is particularly suitable considering real time frame processing, multiple reference frames and fractional pel accuracy.

A quarter pixel full search variable block motion estimation architecture has been proposed that can process all the required motion vectors for H.264/AVC encoder in parallel. Experimental results have shown that the architecture can process in real time upto 5 reference frames at a clock speed of 120MHz[10]. The QP ME hardwares for other block sizes are similar to this hardware. For each 4x4 block in a MB, first, HP ME hardware finds the best HP MV by performing . Half-Pel and Quarter-Pel Search Locations as shown in Figure 2. Half-pel interpolation (HPI) and half-pel search (HPS) and sends this HP MV to QP ME hardware. Then, QP ME hardware finds the best QP MV for that 4x4 block by performing quarter-pel search (QPS) around the location pointed by this HP MV with a search range of [-1, 1]. As the HP ME hardware is performing HPI and HPS, the integer and half pixels necessary for QP accurate ME are send to the search window register file (SWRF) by the HP ME hardware. The proposed layout of the integer and half pixels in the 4x4 SWRF, when the location pointed by the best integer-pel MV is location 17, is shown Figure 3. Since the HP ME will be performed at the HPS locations 8, 9, 10, 16, 18, 24, 25 and 26, the best HP MV will point to one of these locations and the QP ME will be performed at the eight QPS locations around that location. For example, if the best HP MV points to location 8, QP ME will be performed at the QPS locations 8_1, 8_2, 8_3, 8_4, 8_5, 8_6, 8_7 and 8_8.

The control unit sends the read addresses to SWRF based on the best HP MV for accessing the necessary integer and half pixels. Since there are eight HPS locations and there are eight QPS locations for each HPS location, the control unit must beable to generate read addresses for 64 QPS locations (8_1, 8_2, 8_3, … , 26_6, 26_7, 26_8). The QPI datapaths generate the quarter pixels and send them to processing elements (PE). The proposed layout of the integer and half pixels in the 4x4 SWRF provide a good correlation between the read addresses of 64 QPS locations. The read address correlations of 64 QPS locations are shown in Figure 4. Therefore, the control unit generates the read addresses of 64 QPS locations by using

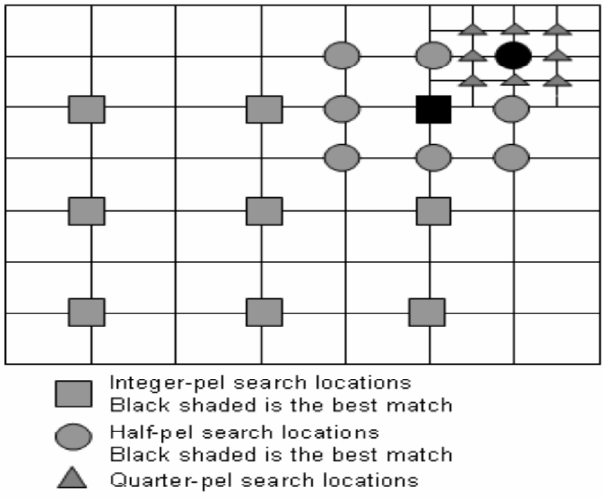

Integer-pel search locations
Black shaded is the best match
Half-pel search locations
Black shaded is the best match
Quarter-pel search locations

Figure 2: Half-pel and Quarter Pel search locations

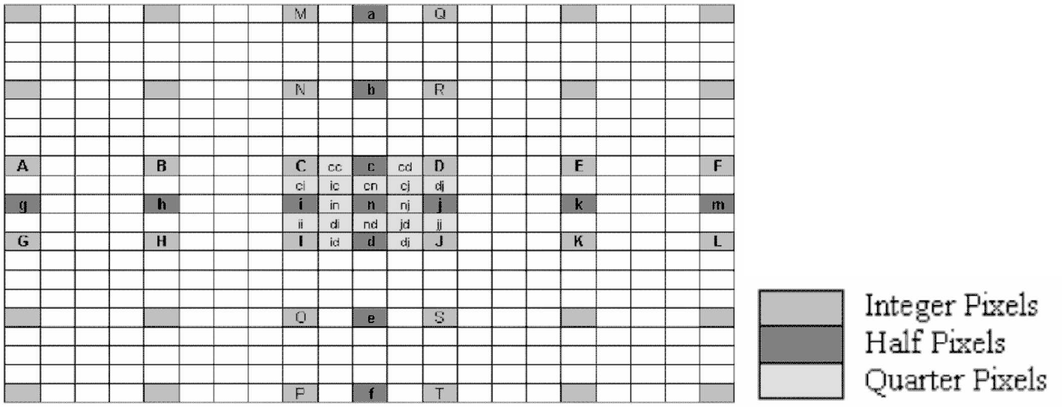
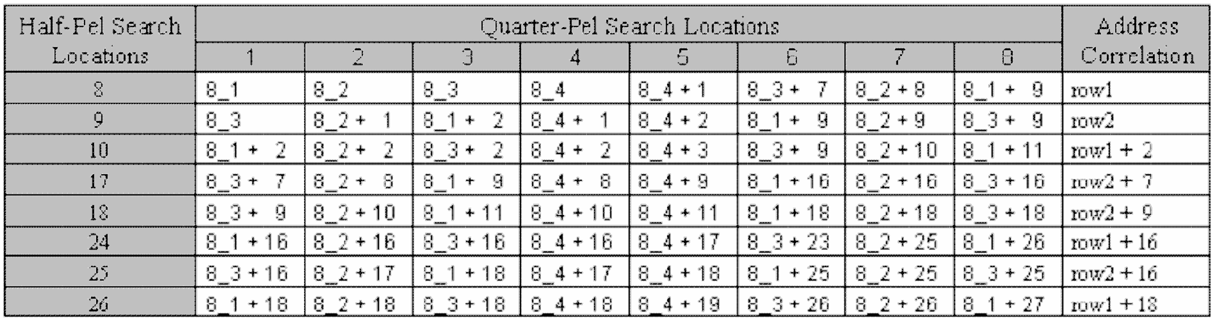


Figure 3: Procedure for Half-Pel and Quarter Pel Interpolation



Half-Pel Search Locations	Quarter-Pel Search Locations								Address Correlation
	1	2	3	4	5	6	7	8	
8	8_1	8_2	8_3	8_4	8_4 + 1	8_3 + 7	8_2 + 8	8_1 + 9	row1
9	8_3	8_2 + 1	8_1 + 2	8_4 + 1	8_4 + 2	8_1 + 9	8_2 + 9	8_3 + 9	row2
10	8_1 + 2	8_2 + 2	8_3 + 2	8_4 + 2	8_4 + 3	8_3 + 9	8_2 + 10	8_1 + 11	row1 + 2
17	8_3 + 7	8_2 + 8	8_1 + 9	8_4 + 8	8_4 + 9	8_1 + 16	8_2 + 16	8_3 + 16	row2 + 7
18	8_3 + 9	8_2 + 10	8_1 + 11	8_4 + 10	8_4 + 11	8_1 + 18	8_2 + 18	8_3 + 18	row2 + 9
24	8_1 + 16	8_2 + 16	8_3 + 16	8_4 + 16	8_4 + 17	8_3 + 23	8_2 + 25	8_1 + 26	row1 + 16
25	8_3 + 16	8_2 + 17	8_1 + 18	8_4 + 17	8_4 + 18	8_1 + 25	8_2 + 25	8_3 + 25	row2 + 16
26	8_1 + 18	8_2 + 18	8_3 + 18	8_4 + 18	8_4 + 19	8_3 + 26	8_2 + 26	8_1 + 27	row1 + 18

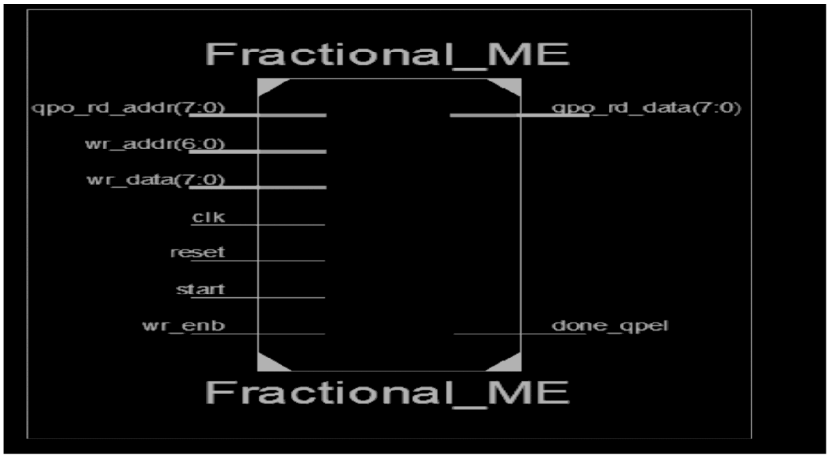
Figure 4: Address correlation of quarter pel search locations

the read addresses of the QPS locations 8_1, 8_2, 8_3, 8_4 and the read address correlations of 64 QPS locations.

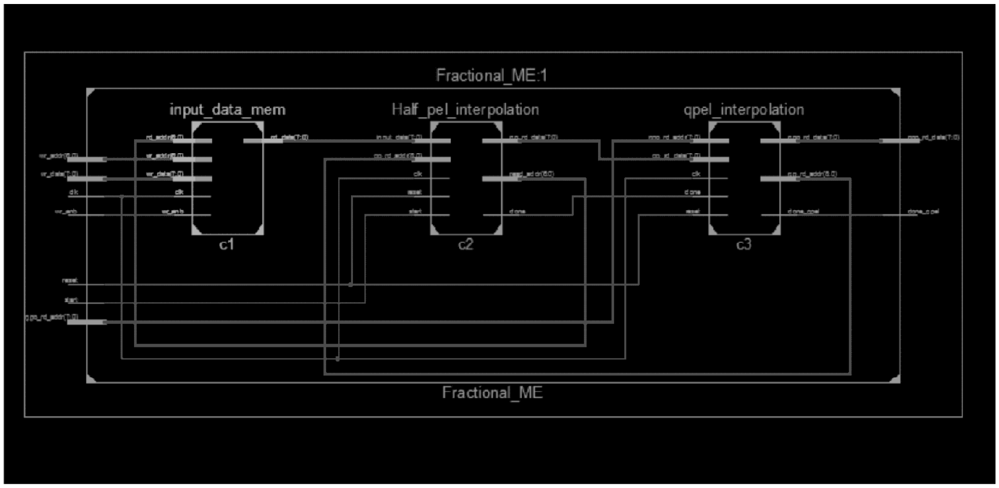
5. PROPOSED ARCHITECTURE

An efficient hardware architecture module with the pixel interpolation is required for H.264 system.The proposed approach is half-pel followed by quarter pel interpolation hardware architecture.Also the proposed architecture performs with high speed and low power which is expected in the video compression for HDTV.

6. SYNTHESIS RESULTS

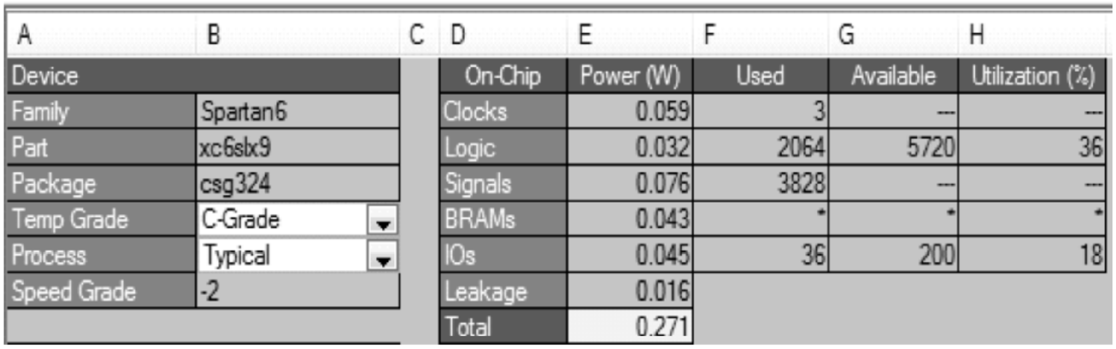


(a)



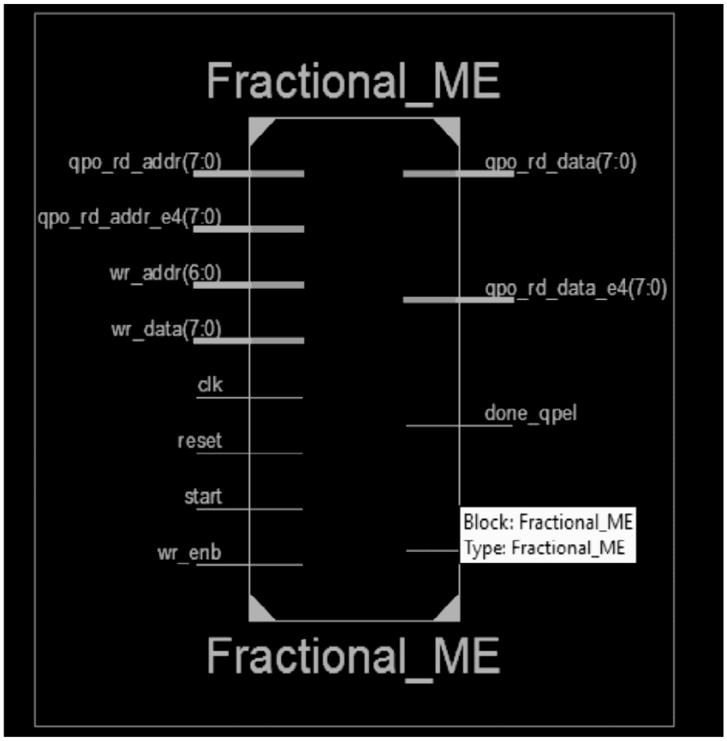
(b)

Figure 5: RTL schematic of Engine I Interpolation module a. Top module b. Detail view

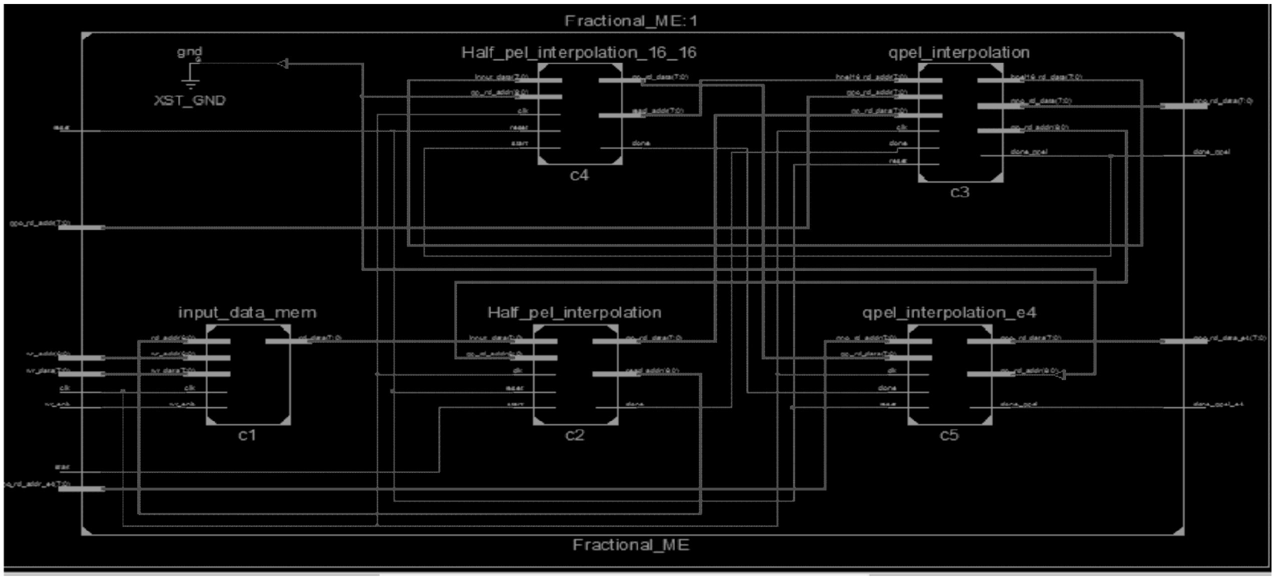


A	B	C	D	E	F	G	H
Device			On-Chip	Power (W)	Used	Available	Utilization (%)
Family	Spartan6		Clocks	0.059	3	---	---
Part	xc6slx9		Logic	0.032	2064	5720	36
Package	csg324		Signals	0.076	3828	---	---
Temp Grade	C-Grade		BRAMs	0.043	*	*	*
Process	Typical		IOs	0.045	36	200	18
Speed Grade	-2		Leakage	0.016			
			Total	0.271			

Figure 6: Power report of engine I

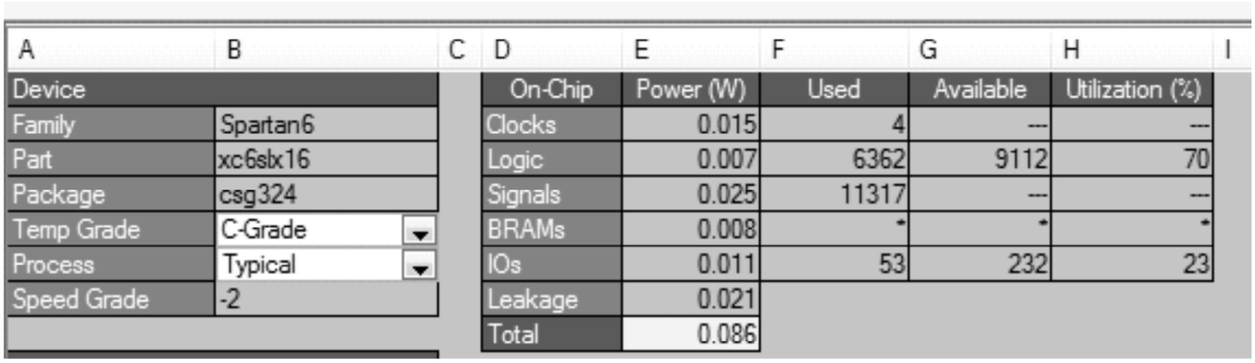


(a)



(b)

Figure 6: RTL schematic of Engine II Interpolation module a.Top module b.Detail view



A	B
Device	
Family	Spartan6
Part	xc6slx16
Package	csg324
Temp Grade	C-Grade
Process	Typical
Speed Grade	-2

D	E	F	G	H
On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.015	4	---	---
Logic	0.007	6362	9112	70
Signals	0.025	11317	---	---
BRAMs	0.008	*	*	*
IOs	0.011	53	232	23
Leakage	0.021			
Total	0.086			

Figure 7: Power report of Engine II

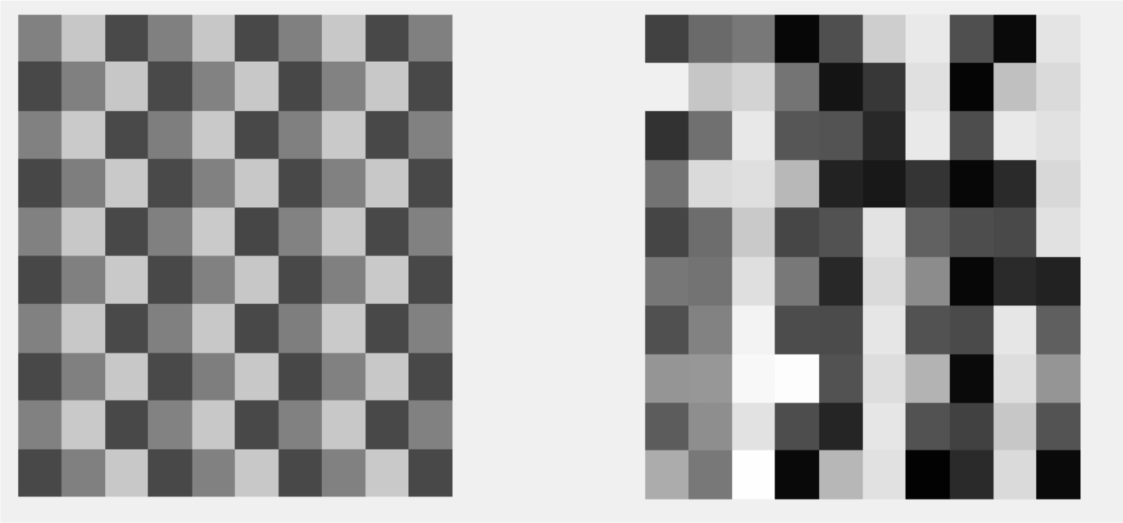


Figure 8: Input and processed output image of engine I

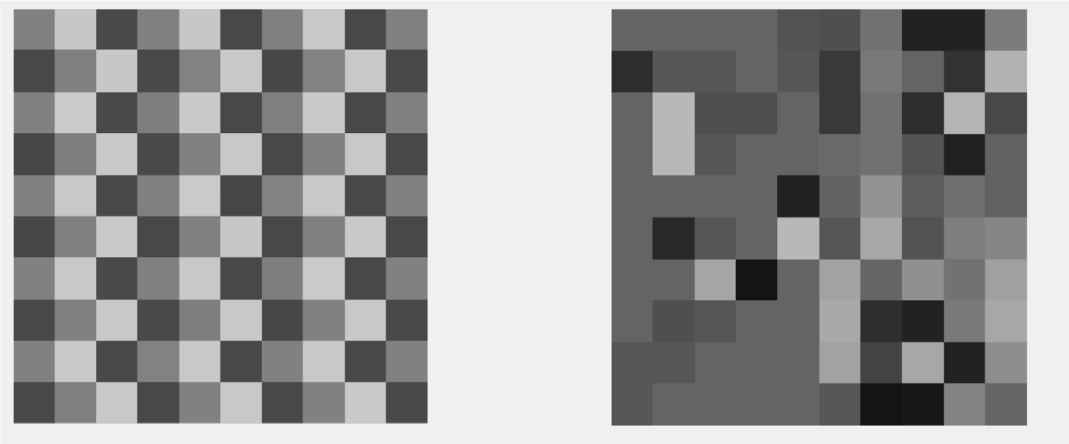


Figure 9: Input and processed output image of engine II

7. COMPARISON WITH PREVIOUS WORK:

	[10]	*[11]*	*[12]*	*Proposed method*
Slices	14.5 K	-	-	6107
LUT‘s	28.5K	-	-	6362
Gate count	225K	321K	448K	271808k
Speed	149.2			
FME Power	-	374mw	135.02mw	0.086w
Cycles/Pixel	-	2.46	1.32	

8. CONCLUSION

In this paper, an efficient VLSI architecture for half pel and quarter pel interpolation for a single frame with two engines are designed. This architecture designed consumes low power and area which can be used as part of a complete H.264 video coding system for portable applications. The proposed hardware architecture is implemented in Verilog HDL. The Verilog RTL code is verified to work at 200 MHz in a Xilinx SpartranVI FPGA.

REFERENCES

[1] G. J. Sullivan, T. Wiegand, and H. Schwarz, “Editors’ draft revision to ITU-T Rec. H.264 | ISO/IEC 14496-10 Advanced Video Coding,” JVT of ISO/IEC MPEG & ITU-T VCEG, JVT-AD205, Feb. 2009.

[2] Y. Vatis, B. Edler, D.N. Nguyen, and J. Ostermann, “Two-dimensional non-separable adaptive Wiener interpolation filter for H.264/AVC,” ITU-T SG16/Q6, Z17, April 2005.

[3] S. Wittmann and T. Wedi, “Separable adaptive interpolation filter,” ITU-T SG16/Q6, C-0219, Geneva, Switzerland, July 2007.

[4] Yang, C., Goto, S., Ikenaga, T.: High performance VLSI architecture of fractional motion estimation in H.264 for HDTV. In Proceedings of the IEEE ISCAS, pp. 2605–2608, Greece (2006).

[5] Chen, Y.H., Chen, T.C., Chien, S.Y., Huang, Y.W., Chen, L.G.: VLSI architecture design of fractional motion estimation for H.264/AVC. J. Signal Process. Syst. 53(3), 335–347 (2008).

[6] Song, Y., Liu, Z., Ikenaga, T., Goto, S.: A VLSI architecture for variable block size motion estimation in H.264/AVC with low cost memory organization. IEICE Trans. Fundam. E89(12), 3594–33].

[7] “Fast integer pel and fractional pel motion estimation for AVC,” in Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-F016, December 2002.601 (2006).

[8] Y. W. Huang *et al.*, “Hardware architecture design for variable block size motion estimation in MPEG-4 AVC/JVT/ITU-T H.264,” Proceedings of the 2003International Symposium on CAS, ISCAS ’03, pp. II-796-II-799, May 2003.

[9] S. Y. Yap and J. V. Mc Canny, “A VLSI architecture for variable block size video motion estimation,” IEEE Transactions on CAS II, vol. 51, no. 7, July 2004.

[10] Choudhury A. Rahman and WaelBadawy,”A Quarter Pel Full Search Block Motion Estimation Architecture For H.264/ Avc

[11] C.-Y. Kao, C.-L. Wu, Y.-L. Lin, “A High-Performance Three-Engine Architecture for H.264/AVC Fractional Motion Estimation”, IEEE Trans. VLSI Syst., vol.18, No. 4, pp. 662-666, April, 2010.

[12] P.-K. Tsung, W.-Y. Chen, L.-F. Ding, C.-Y. Tsai, T.-D. Chuang, and L.-G. Chen, “Single-iteration full-search fractional motion estimation for quad full HD H.264/AVC encoding,” in Proc. ICME, 2009, pp. 9–12.