Parallel Processing Deblocking Filter Hardware for High Efficiency Video Coding

Darshan G V[1], Dinesha P[2]

[1] *(Department of ECE, Dayananda Sagar College of Engineering, India, itsdarshangv@gmail.com)*
[2]*(Department of ECE, Dayananda Sagar College of Engineering, India, dineshprg@gmail.com)*

ABSTRACT: High Efficiency Video Coding (HEVC) is a recent video compression standard, it uses adaptive deblocking filter for reducing blocking artifacts. This deblocking filter performs strong filtering, weak filtering or no filtering based on different conditions. HEVC deblocking filter (DBF) have high computational complexity but it provides about 50% higher bit rate reduction for same perceptual video quality compared to H.264. This HEVC deblocking filter is used in both encoder and decoder for HEVC. Deblocking filter improves the overall quality of video frame by improving both subjective and objective quality. Using parallel processing/pipelining the efficiency of the system could be improved. The proposed work implements parallel datapath system and thus enhances the throughput. It is implemented using Verilog HDL and the FPGA used is Xilinx vertex-5

KEYWORDS: High Efficiency Video Coding, deblocking filter, blocking artifacts, datapath, throughput

I INTRODUCTION

International Telecommunication Union (ITU) introduced a new H.265 video compression standard for high definition and ultra-high definition applications. H.265, is known as High Efficiency Video Coding (HEVC), more efficient than its predecessor H.264/AVC. The compression performance of H.265/HEVC is twice as high as that of H.264/AVC at the expense of increased computational complexity. Moreover, H.265/HEVC improves the algorithm to easily support parallel processing.The deblocking filter of H.265/HEVC or H.264/AVC plays an important role in the video coding system. Fig. 1 shows the block diagram of H.265/HEVC encoder. The deblocking filter reduces the blocking artifacts occurring from the quantization of transform coefficients and the block-based motion compensation. The data dependency in H.264/AVC limits the parallelism of the design. H.265/HEVC improves the parallelism by removing the data dependency between the adjacent filtering operations.

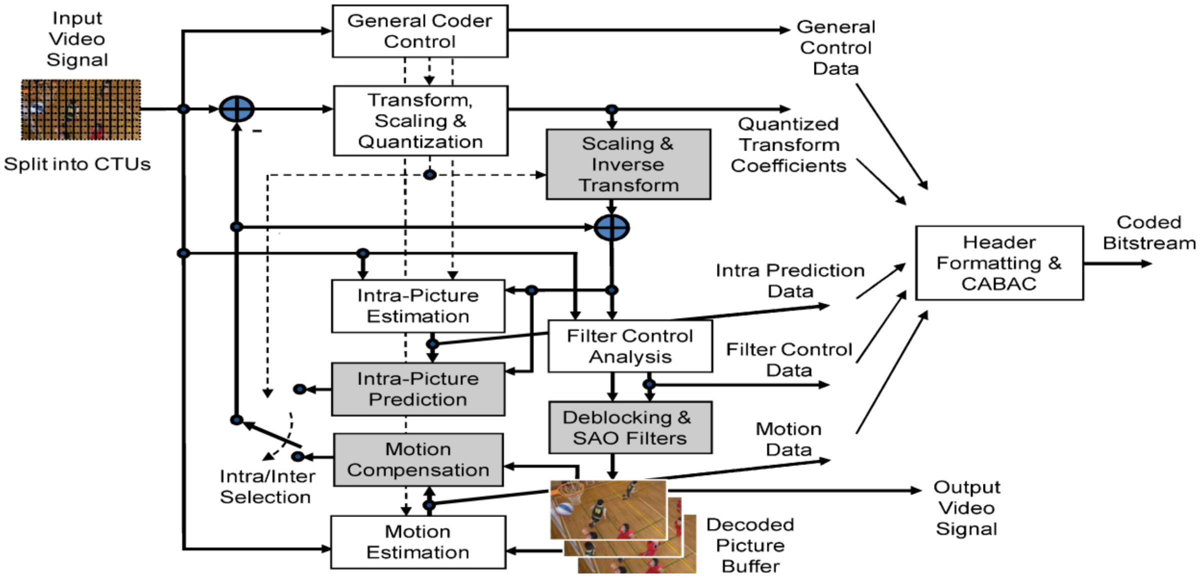


Figure 1: Block diagram of HEVC encoder

The aim is to implement a high throughput HEVC deblocking filter. So, parallel datapaths should be implemented for improved throughput eventhough the complexity increases. The Deblocking filter should be implemented such that is efficient. Through parallelism/pipelining the system throughput will be improved. Verilog HDL is used to model digital systems and designing and verification of their functionality. This will be used to implement the deblocking filter on FPGA. Verilog HDL is simple and similar to the C programming language, which implements its design in terms of hierarchy of modules.

There are many proposals on deblocking filter already available, High Efficiency Video Coding (HEVC) international video compression standard uses adaptive deblocking filter for reducing blocking artifacts. Multiple datapaths are used to improve its performance [1].The overall computation is pipelined and a new parallel-zigzag processing order is introduced to achieve high throughput. The processing order of the filter is efficiently rearranged to process both horizontal edges and vertical edges at the same time. This H.265/HEVC deblocking filter architecture improves the parallelism by dissolving the data dependency between the adjacent filtering operations [2].As the next generation standard of video coding, the High Efficiency Video Coding (HEVC) aims to provide significantly improved compression performance in comparison with all other existing video coding standards. This work uses a four-stage pipeline hardware architecture on a quarter-LCU basis of deblocking filter in HEVC and a memory interlacing technique is used to increase the throughput [3].The in-loop deblocking filter used in the High Efficiency Video Coding (HEVC) standard to reduce visible artifacts at block boundaries. Compared to the H.264/AVC deblocking filter, the HEVC deblocking filter has lower computational complexity and better parallel processing capabilities while still achieving significant reduction of the visual artifacts [4].The new deblocking filter (DF) tool of the next generation HEVC standard is one of the most time consuming algorithms in video decoding. In order to achieve real-time performance at low-power consumption, a hardware accelerator for this filter is developed [5].

II HEVC DEBLOCKING FILTER

HEVC deblocking filter is an adaptive filter which removes the blocking artifacts present by applying suitable conditions and improve the subjective and objective quality.Using parallal datapaths the speed of deblocking filter operation can be improved, thereby improving the overall throughput of the overall system. Deblocking filter is a very useful block in both encoder and decoder of HEVC, so a high throughput deblocking filter is necessary with increasing quality and size of video.The main difficulty while designing a deblocking filter is to decide whether the filtering process will be applied or not for a particular block boundary, as well as to decide the strength of the filtering to be applied. Two opposite cases can happen if the filtering decisions are not well designed. On one hand, excessive filtering may lead to an excessive smoothing of the picture details, causing loss of data and reduction of the subjective quality. On the other hand, lack of filtering may lead to blocking artifacts that are easily identified by the human visual system, resulting in decrease of video subjective quality. This later problem is precisely the problem that the filter is designed to deal with.

The deblocking filter in HEVC has been designed to improve the subjective quality while reducing the computational complexity, when compared to deblocking filter of the H.264 standard. Furthermore, the HEVC the deblocking filter is more suitable for parallelization when compared its corresponding in H.264 standard, since it is designed in a way to prevent spatial dependencies across the picture.

For luma components, only block boundaries with BS equal to one or two are filtered. This implies that usually no filtering is applied in flat areas of the image. It helps to avoid multiple subsequent filtering in areas of the image where samples are copied from one part to another with a residual equals to zero, which could lead to an unnecessary over-smoothing of the area. The deblocking filter is first applied to all sets of two 4x4 neighboring blocks and share a vertical boundary. After all vertical block boundaries have been filtered, all blocks belonging to the frame sharing a horizontal boundary are then tested and filtered. For chroma components, only block boundaries with BS equal to 2 are subject to filtering operations. Therefore, only blocks boundaries containing at least one intra block are filtered.

III. IMPLEMENTATION

A deblocking filter for HEVC/H.265 is implemented using Verilog HDL and MATLAB for conversion of image into text and vice-versa. The implementation flow of deblocking filter is as shown in fig 2.

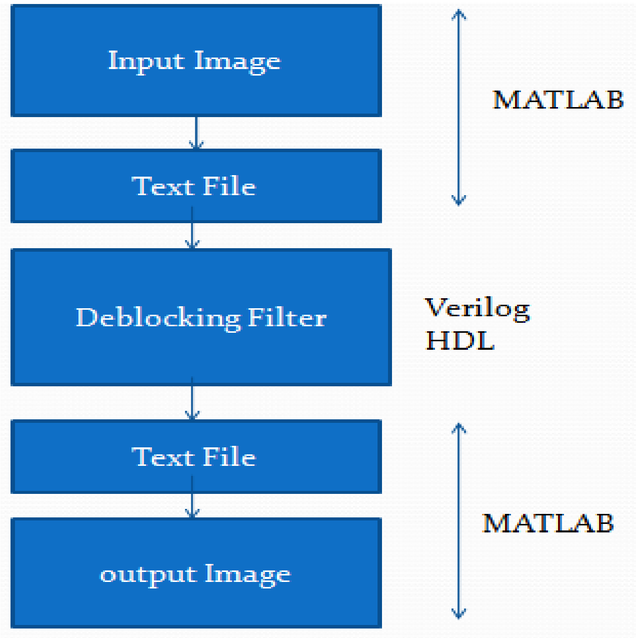
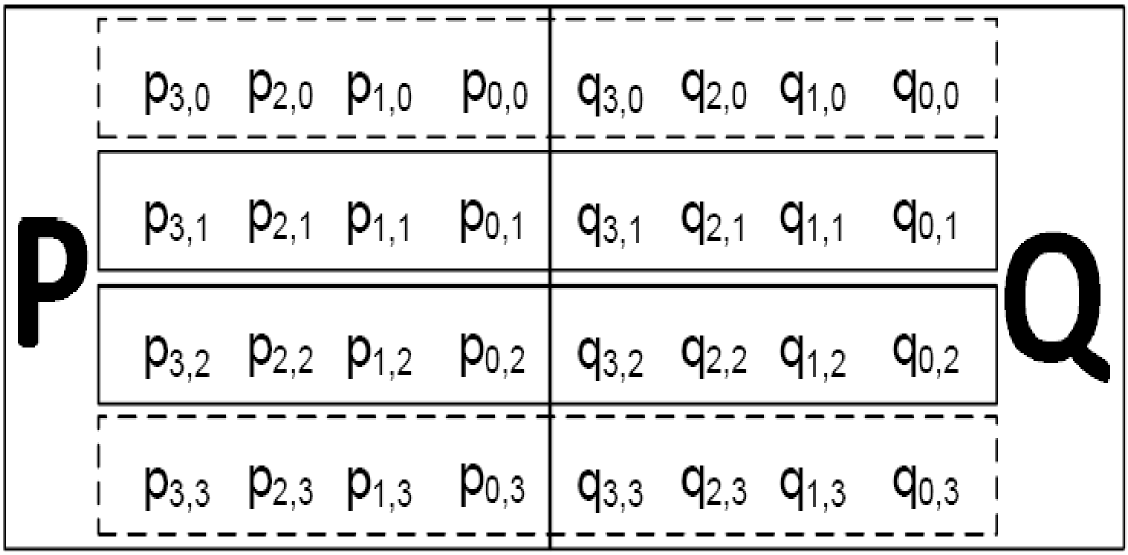


Figure 2: Implementation flow

Filtering operation is applied for the edge between CU P and CU Q. The deblocking decisions are made based on two adjacent blocks as shown in Fig 3,which shows samples from an 8x8 block.



$p_{3,0}$	$p_{2,0}$	$p_{1,0}$	$p_{0,0}$	$q_{3,0}$	$q_{2,0}$	$q_{1,0}$	$q_{0,0}$	
$p_{3,1}$	$p_{2,1}$	$p_{1,1}$	$p_{0,1}$	$q_{3,1}$	$q_{2,1}$	$q_{1,1}$	$q_{0,1}$	
$p_{3,2}$	$p_{2,2}$	$p_{1,2}$	$p_{0,2}$	$q_{3,2}$	$q_{2,2}$	$q_{1,2}$	$q_{0,2}$	
$p_{3,3}$	$p_{2,3}$	$p_{1,3}$	$p_{0,3}$	$q_{3,3}$	$q_{2,3}$	$q_{1,3}$	$q_{0,3}$	

Figure 3: Two adjacent blocks P and Q

Module filter will decide whether to filter or not and the filter mode operation (strong filter or weak filter). First of all, value of BS must be checked, if BS =0 no filtering operation is performed. When BS> 0, the following condition is checked to decide whether the deblocking filter is applied or not [6]:

| p2,0 – 2*p1,0 + p0,0 |+| p2,3 – 2*p1,3 + p0,3 |+| q2,0 – 2*q1,0 + q0,0 |+ | q2,3 – 2*q1,3 + q0,3 | < β
(1)

where the threshold β depends on QP. Only the first and the fourth lines in a block boundary of length 4 are evaluated to avoid unnecessary complexity.

If condition (1) is true, the filter is applied to the edge, otherwise not. Whether to apply a strong or weak De-blocking is also determined based on the first and the fourth lines across the block boundary of four samples. The following conditions are investigated to determine the filter mode operation:

| p2,0 – 2*p1,0 + p0,0 |+| q2,0 – 2*q1,0 + q0,0 | < β/8
(2)

| p2,3 – 2*p1,3 + p0,3 |+| q2,3 – 2*q1,3 + q0,3 | < β/8
(3)

$| p3,0 – p0,0 | + | q0,0 – q3,3 | < \beta/8$
(4)
$| p3,3 – p0,3 | + | q0,3 – q3,3 | < \beta/8$
(5)
$| p0,0 – q0,0 | < 5tc/2$
(6)
$| p0,3 – q0,3 | < 5tc/2$
(7)

If these conditions are true, the strong filter is selected. Then p0 to p2 and q0 to q2 are the filtered pixels, and the filter operations are as follows:
$p0' = (p2 + 2*p1 + 2*p0 + 2*q0 + q1 + 4) >> 3$
$q0' = (p1 + 2*p0 + 2*q0 + 2*q1 + q2 + 4) >> 3$
$p1' = (p2 + p1 + p0 + q0 + 2) >> 2$
$q1' = (p0 + q0 + q1 + q2 + 2) >> 2$
$p2' = (2*p3 + 3*p2 + p1 + p0 + q0 + 4) >> 3$
$q2' = (p0 + q0 + q1 + 3*q2 + 2*q3 + 4) >> 3$
Otherwise the weak filter is selected. Then there are three conditions is investigated:
$| p2,0 – 2*p1,0 + p0,0 |+| p2,3 – 2*p1,3 + p0,3 | < 3\beta/16$
(8)
$| q2,0 – 2*q1,0 + q0,0 |+| q2,3 – 2*q1,3 + q0,3 | < 3\beta/16$
(9)
$| \Delta0,I | <10tc \ (0<= i<=3)$
(10)
If condition (10) is true the modified value p0' and q0' are calculated as follows:
$p0' = p0 +\Delta0$
$q0' = q0 –\Delta0$

If condition (8) holds the modified p1' is calculated as follows:
$Dp = Clip3(-(tC >> 1), tC >> 1, (((p2 + p0 + 1) >> 1) – p1 + D) >>1)$
$p1' = Clip1Y(p1 + Dp)$
Likewise, if condition (9) holds, q1' is calculated as:
$Dq = Clip3(-(tC >> 1), tC >> 1, (((q2 + q0 + 1) >> 1) – q1 – D) >>1)$
$q1'=Clip1Y(q1-Dq)$
(11)

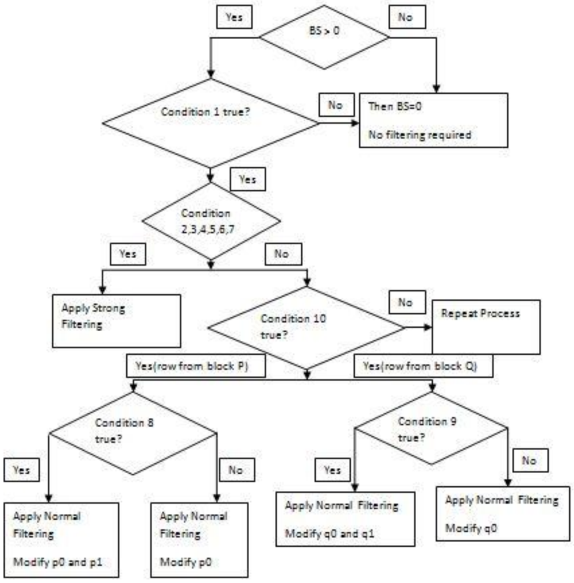


Figure 4: Flow of deblocking filter operation

IV RESULTS

The deblocking filter for HEVC/H.265 will be implemented using Verilog HDL. At 100MHz clock frequency throughput of 0.8, 1.6 and 3.2Gbps are obtained for 1,2 and 4 datapaths respectively. This filter is thus capable of operating on high resolution video.

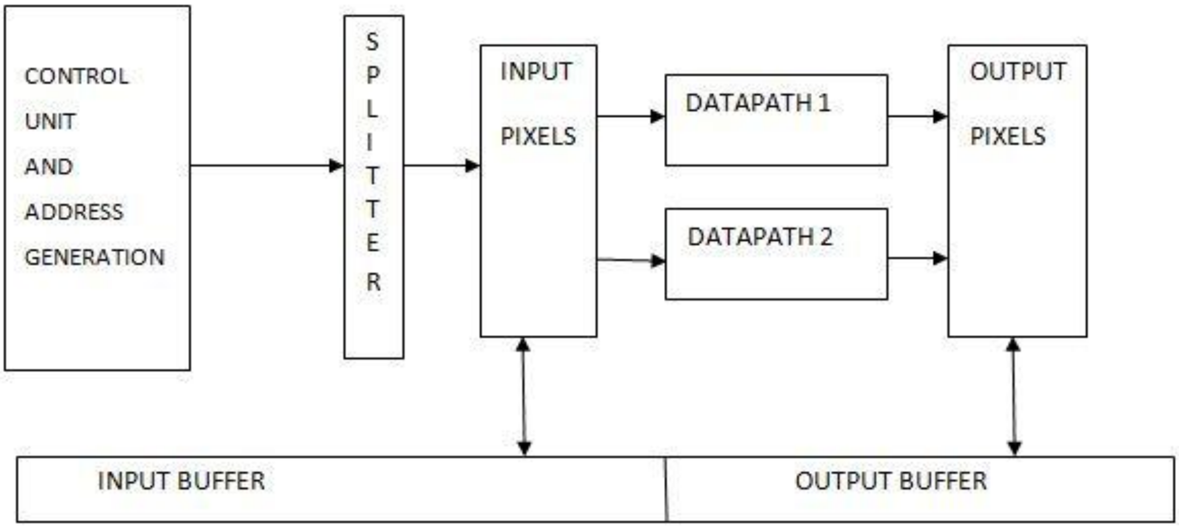
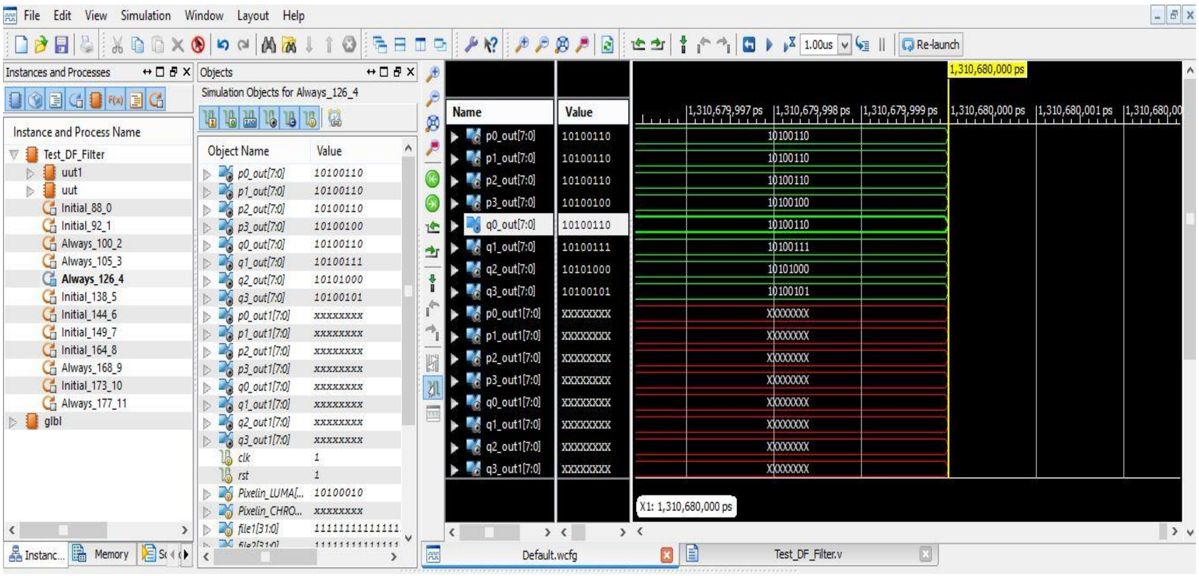


Figure 5: HEVC DBF hardware

Table I :Throughput for different number of deblocking filter in parallel

Number of Datapaths(Deblocking Filter)	***Execution Time(µs)***	***Throughput(Gbps)***
1	***655.34***	***0.8***
2	***327.67***	***1.6***
4	***163.54***	***3.2***

To this deblocking filter when images are given as input it removes the blocking artifacts present in the image as shown in Fig 7. The deblocking filter is implemented in hardware using Verilog HDL and a part of the detailed RTL for deblocking filter hardware is shown in Fig 8 when it is implented on a vertex 5 FPGA.



Object Name	Value
p0_out[7:0]	10100110
p1_out[7:0]	10100110
p2_out[7:0]	10100110
p3_out[7:0]	10100100
q0_out[7:0]	10100110
q1_out[7:0]	10100111
q2_out[7:0]	10101000
q3_out[7:0]	10100101
p0_out1[7:0]	xxxxxxxx
p1_out1[7:0]	xxxxxxxx
p2_out1[7:0]	xxxxxxxx
p3_out1[7:0]	xxxxxxxx
q0_out1[7:0]	xxxxxxxx
q1_out1[7:0]	xxxxxxxx
q2_out1[7:0]	xxxxxxxx
q3_out1[7:0]	xxxxxxxx
clk	1
rst	1
Pixelin_LUMA[...	10100010
Pixelin_CHRO...	xxxxxxxx
file1[31:0]	11111111111111

Figure 6: Simulation result of deblocking filter.

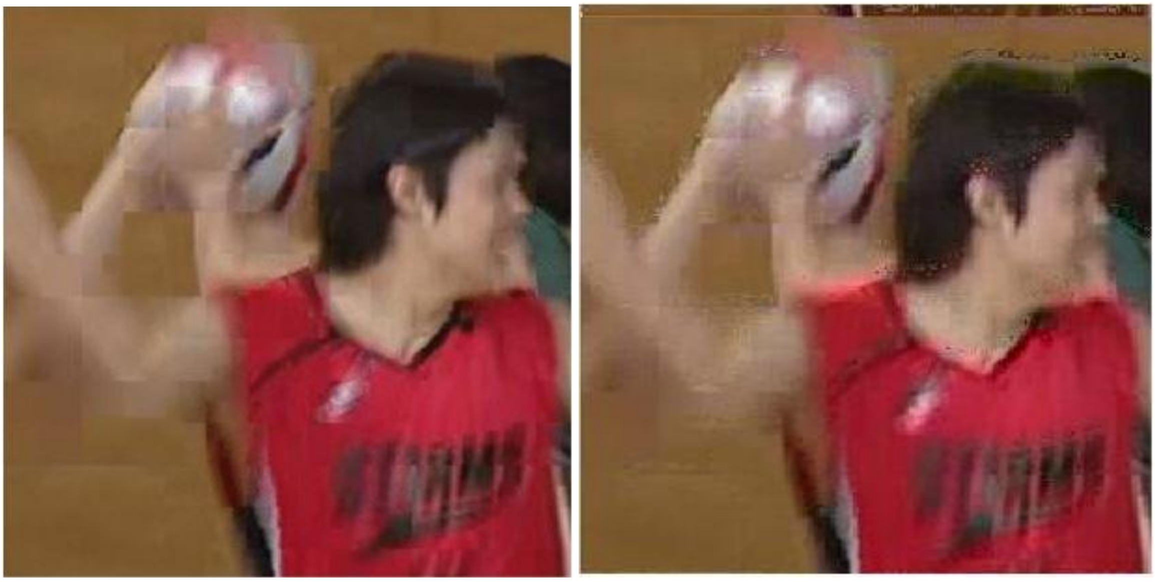


Figure: 7 a) Input Image b)Output image of deblocking filter

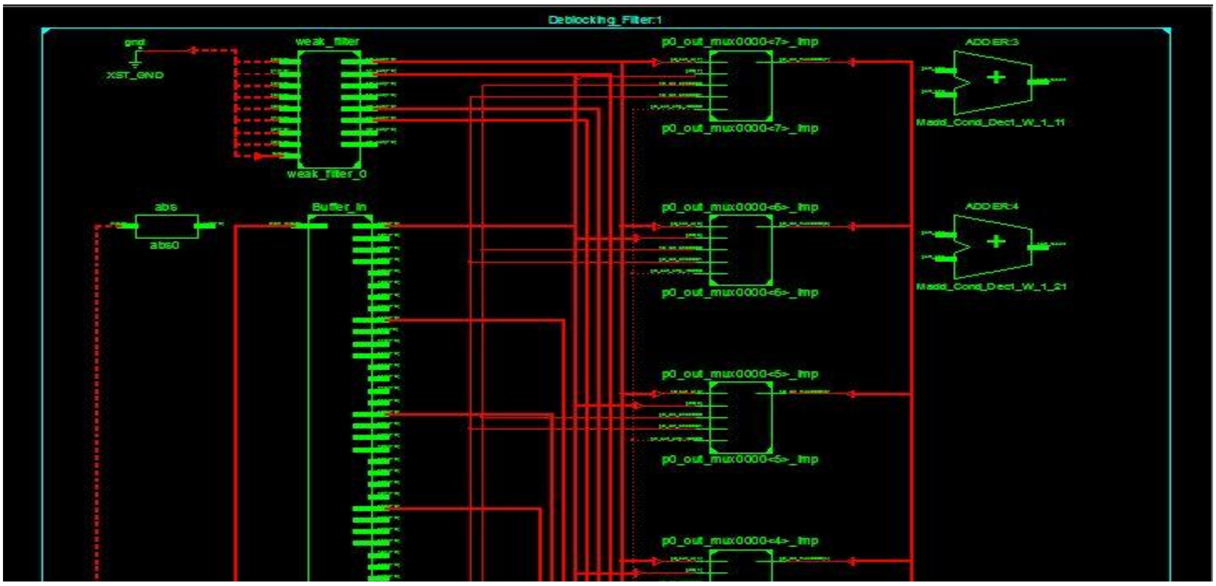


Figure 8: Detailed RTL of a part of deblocking filter hardware.

V .CONCLUSION

In this Paper, MATLAB implementation of image to text and text to image is realized and HEVC deblocking filter is implemented using Verilog code. HEVC deblocking filter, which removes blocking artifacts in an image is implemented for high throughput using parallel datapaths. Using parallel/pipeling the performance of any system can be increased due to improvement of speed and throughput. This deblocking filter could be used in both real time HEVC encoder and decoder. Here a high speed deblocking filter is implemented which will be useful in HEVC.

REFERENCES

[1] E. Ozcan, Y. Adibelli, and I. Hamzaoglu, A high performance deblocking filter hardware for high efficiency video coding, *IEEE Trans. Consumer Electron*, vol. 59, no. 3, 2013, 714–720.

[2] Hoai-Huong Nguyen Le and Jongwoo Bae, High-Throughput Parallel Architecture for H.265/HEVC Deblocking Filter, *Journal Of Information Science And Engineering*, 30, 2014, 281-294.

[3] W. Shen, Q. Shang, S. Shen, Y. Fan, and X. Zen, A high-throughput VLSI architecture for deblocking filter in HEVC, in *Proc. IEEE Int.Symp. Circuits Syst.*, 2013, 673–676.

[4] A. Norkin *et al.*, HEVC deblocking filter, *IEEE Trans. Circuits Syst.Video Technol.*, vol. 22, no. 12, 2012, 1746–1754.

[5] *Cláudio Machado Diniz, Muhammad Shafique, Felipe Vogel Dalcin, Sergio Bampi, Jörg Henkel* , A Deblocking Filter Hardware Architecture for the High Efficiency Video Coding Standard, *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2015, 1509-1514.

[6] Vivienne Sze , Madhukar Budagavi and Gary J. Sullivan, High *Efficiency Video Coding (HEVC) Algorithms and Architectures(*Springer 2014).