

A High Speed Vlsi Architecture For Image Deinterleaver For Compression

peddi Surekha & ambati Sridevi

Assistant Professor. E.C.E Department Chebrolu Engineering College Chebrolu, Guntur, A.P. India
M. Tech – Scholar E.C.E Department Chebrolu Engineering College Chebrolu, Guntur, A.P. India

ABSTRACT: *The community of the research in the last few years has received significant attention from the field of approximate computing, particularly in different signal processing context. The compression algorithms of the image deinterleaver and video such as MPEG and JPEG and so on, which can be exploited to realize the implementations of highly power-efficient of these algorithms.*

However, existing approximate architectures naturally fix the approximations of hardware levels statically and are not adaptive to the input data. This project addresses this issue through proposing a reconfigurable approximate for encoders of MPEG which optimizes consumption of power with the aim of maintaining a particular peak signal-to-noise ratio threshold for any video. I design reconfigurable adder/ subtract blocks, and later integrate these blocks in the special levels all the video in to image deinterleavers these image deinterleavers will be converted to digital form and then compression of the image deinterleaver.

I. INTRODUCTION

Introducing a limited amount of computing imprecision in image deinterleaver and video dispensation algorithm separately. The results in an insignificant amount of perceptible visual Change in the output, which makes these algorithms as ideal candidates for the use of

Approximate Computing architectures develop the fact that a small relaxation in the output correctness can result in considerably simpler and lower power implementations. On the other hand, the most approximate hardware architectures projected so far suffer from the restriction that, for widely varying input parameters, it becomes very tough to provide a quality bound on the output, and in some other cases, the output quality may be severely degraded. The main reason for this output quality variation is that the degree of approximation (DA) in the hardware architecture is permanent statically and cannot be modified for different inputs. One possible solution is to assume a conservative approach and use a very low DA in the hardware so that the output accuracy is not drastically exaggerated. However, such a conventional approach will, as expected, considerably impact the power savings as well.

II. EXISTED SYSTEM

There are of two compression techniques. One is lossless compression and another one lossy compression. Lossy compression is employed to decrease the data of the image deinterleaver, picture or video. By victimization of this compression technique the receiver could lose some data. It will be loss to the receiver as well as to the sender. In the applications of net and transmission lossy compression is the common usage.

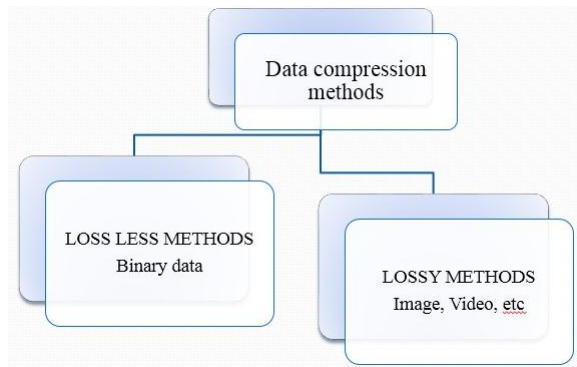


FIG. 1 BLOCK DIAGRAM

To overcome this drawback of reduction of the information or data, a brand new compression technique was been introduced, i.e., lossless compression. Lossless compression is exactly opposite to lossy compression. Lossy compression is irreversible and lossless compression is reversible compression. Lossless compression sends the image deinterleaver without reduction of the information to the receiver.

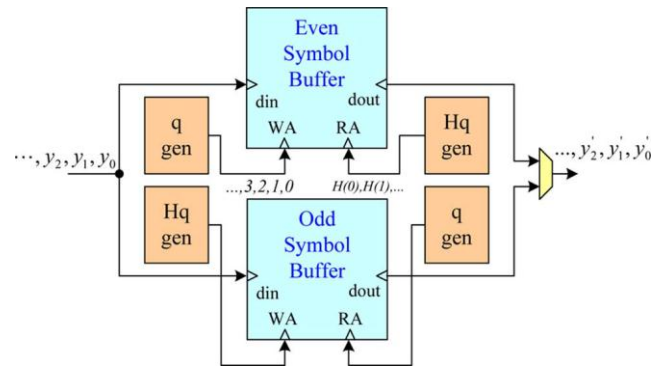


FIG. 2 BLOCK DIAGRAM OF THE TYPICAL SYMBOL-DEINTERLEAVER DESIGN.

The above block diagram consists of two buffers known as ping-pong symbol buffers. In these two buffers one is even symbol buffer and another is odd symbol buffer. Each buffer can be recognized by an On-chip SRAM block which consists of N_{\max} cells. The functioning of each buffer is to hold the entire incoming symbol of data.

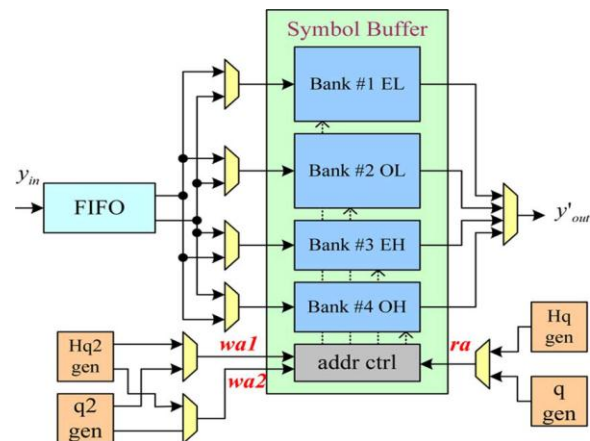


FIG. 3 Overall architecture of the existed Deinterleaver design.
The be different for even and odd symbols. These two buffers can be operated under the

control of the associated address generators. These address generators are q generator and Hq generator. These address generators are used for the control of data accesses. The buffering operations for even and odd symbols are different. For each buffer either memory read or memory write operation will takes place at any time and the port number can be only one for the On-chip SRAM. The size of each memory cell is equal to the word length of the deinterleaved data. Due to the noise occurred during transmission, the receiver will sample extra bits for better recovery of the data.

III. PROPOSED SYSTEM

The overall architecture of the proposed DVB symbol deinterleaver is shown in FIG 4. As we know that symbol buffer can be used for storing the incoming symbol of data. The proposed deinterleaver design consists of a single symbol buffer. The symbol buffer contains four memory banks. They are 1.EL used for the storage of even-low indexed data 2. OL used for the storage of odd-low indexed data 3. EH used for the storage of even-high indexed data 4. OH used for the storage of odd-high indexed data.

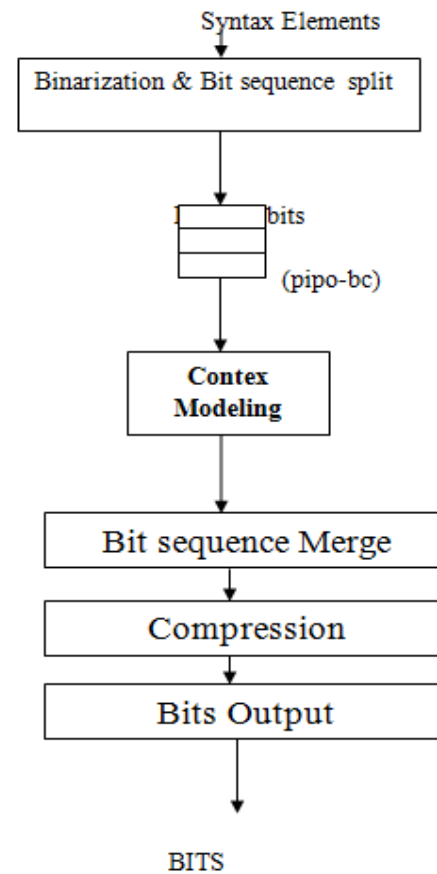


FIG. 4 PROPOSED SYSTEM

The size of EL, OL memory banks is 2048 words and the size of EH, OH memory banks is 1024 words. When buffering the input even symbol, the data will be written to the memory in sequential steps. Therefore, EL and OL memory banks will access for the first $M_{\max}/2$ data and EH, OH memory banks will access the remaining data alternatively. While buffering the input even symbol, the previous odd symbol data which is stored in the buffer will be fetched

out. The memory read and write operations can be done alternatively in two separate memory banks.

The buffering of odd symbol data will be more complex due to written following the permuted order. The division of memory banks according to the even and odd indexed data can be suitable for buffering the even symbol. It is not suitable for odd symbol because there is chances for memory read and write in the same memory bank which cannot be eliminated. In order to solve this problem our memory banks are divided according to the index of the stored data is larger than $(M_{\max}/2)$ or not. The input data which cannot be written to the memory due to the conflict, will temporarily stored in to a first-in- first-output (FIFO).

The main drawback for permutation function generator circuit is that it cannot give the valid address to be generated every cycle because the range of the permuted pseudo numbers may exceed the maximum address bound. If the valid address cannot be generated the data which is entering into the buffer will be stored in temporary registers. This drawback is overcome by using the operation of toggle bit T for the next generated value.

By exploitation lossless compression, the receiver won't lose any information or data from the image deinterleaver picture or video. The quality of the image deinterleaver also won't be modified by the Lossless compression. The process of the lossless compression is: At initial stage the information or the data of the image deinterleaver are remodeled into binary forms i.e., zero and one format (0,1). This information can splits into rows and columns. This can be referred to as binarization. During this method the binary digits can forms like bits in a very sequence. And these bits can forward to down as the regular bits.

These bits are ready to merge. Now the data is united with the primary digit of the primary row with the primary digit of the primary column. During this method all the rows and columns are united.

Then the binary data is prepared to compress. As I know that, are utilizing lossless compression technique here to change the image deinterleaver. The image deinterleaver is reworked into binary data here just in case of lossless compression. This can be very helpful technique to

achieve the accurate image deinterleaver as we would like. In this technique first the binary data will be remodeled into black and white format as we tend to shown in the fig.3. Then the image deinterleavers are going to be transformed into binary data i.e., zero and one (0,1) format as I tend to shown in fig.4. Here I tend to square measure victimization sixty four bit compression that is extremely helpful to the rework. Finally lossless compression can send the original data as output. The ultimate output comes without any loss of information within the image deinterleaver. As a result, we tend to used lossless compression technique for this point. Lossless compression is extremely advantageous and really helpful technique than lossy compression technique.

IV. RESULTS:



FIG: 4 COLOR IMAGE DEINTERLEAVER

The above image deinterleaver (Fig.4) is to be sent to another person. If this is the case, then we need to change the image deinterleaver into black and white image deinterleaver as in Fig.5.



FIG.5 BLACK AND WHITE

This black and white image deinterleaver is further converts to 0, 1 format as it is in the Fig.6.

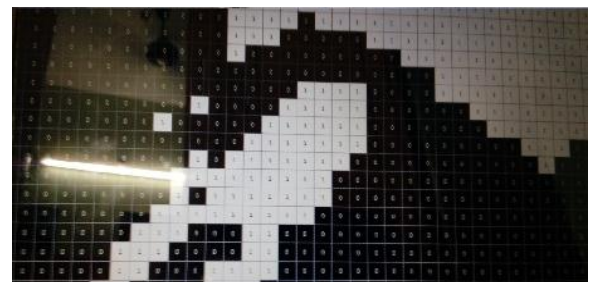


FIG.6.BINARY DATA

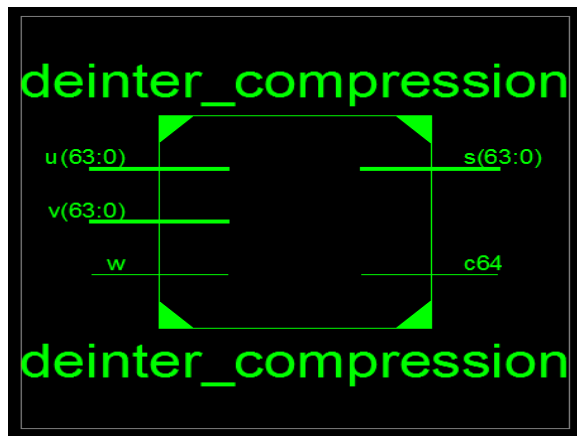


Fig.7 RTL Schematic diagram and the output of compressed data

The above figure shows the RTL Schematic diagram and the output of compressed data is shown in below figure.

Name	Value	1,999,995 ps	1,999,996 ps	1,999,997 ps	1,999,998 ps	1,999,999 ps
u(63:0)	11110000110	111100001100110011001111111100001010101011100000111001010011				
v(63:0)	11001010110	11001010110011110001001100110011001100110011001100110011				
w	2					
s(63:0)	1011011100	10110111001101111100011001000111101101011001001100101111011X				
c64	1					
q(63:0)	11000000110	1100000011001100000111111111000000100010101110000011100000000011				

FIG 8 OUTPUT

V. CONCLUSION

This new and advanced proposed system tells us the about the transformation of the image deinterleaver from one device to another device. Suppose I want to send a colorimage deinterleaver from Transmitting to Rx, then I have to consume more data. To deduce the

data, I want to change the image deinterleaver by following steps. By using this technique, the colourimage deinterleaver will be transformed into black and white image deinterleaver and then it will be transformed into 0, 1 binary form by compressing the data. Then with the help of compression technique data is compressed without loss of data. So, finally the colorimage deinterleaver is converted to binary data with the help of mat lab and the binary data is compressed by using Xilinx 14.7.

VI. REFERENCES:

- [1] F. Ahmed-Zaid, F. Bai, S. Bai, C. Basnayake, B. Bellur, S. Brovold, *et al.*, "Vehicle safety communications—Applications (VSC-A) final report," U.S. Dept. Trans., Nat. Highway Traffic Safety Admin., Washington, DC, USA, Rep. DOT HS 810 591, Sep. 2011.
- [2] J. B. Kenney, "Dedicated short-range Communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.
- [3] J. Daniel, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich, "Design of 5.9 GHz DSRC-based vehicular safety

communication,”*IEEE Wireless Commun. Mag.*, vol. 13, no. 5, pp. 36–43, Oct. 2006.

[4] P. Benabes, A. Gauthier, and J. Oksman, “A Manchester code generator running at 1 GHz,” in *Proc. IEEE, Int. Conf. Electron., Circuits Syst.*, vol. 3, Dec. 2003, pp. 1156–1159.

[5] A. Karagounis, A. Polyzos, B. Kotsos, and N. Assimakis, “A 90nm Manchester code generator with CMOS switches running at 2.4 GHz and 5 GHz,” in *Proc. 16th Int. Conf. Syst., Signals Image deinterleaver Process.*, Jun. 2009, pp. 1–4.

[6] Y.-C. Hung, M.-M. Kuo, C.-K. Tung, and S.-H. Shieh, “High-speed CMOS chip design for Manchester and Miller encoder,” in *Proc. Intell. Inf. Hiding Multimedia Signal Process.*, Sep. 2009, pp. 538–541.

[7] M. A. Khan, M. Sharma, and P. R. Brahmanandha, “FSM based Manchester encoder for UHF RFID tag emulator,” in *Proc. Int. Conf. Comput., Commun. Netw.*, Dec. 2008, pp. 1–6.



AMBATI SRIDEVI studied B.Tech at PBR VITS, Kavali. At present she is pursuing M.Tech at chebrolu engineering college. Her area of interest is Communications.



PEDDI SUREKHA studied B.Tech at VIGNAN Engineering College and M.Tech at Nalanda Engineering College. She has 4 years of teaching experience as assistant professor. Her area of interest is Embedded Systems.