

Reusable High Speed Architecture Design for Video Transmission In Satellite Applications

Arun Kumar P.

School Of Electronics Engineering, VIT University, Vellore-632014, India

Email: arunkumarvlsi@gmail.com

Abstract. This paper presents a reusable technique for high speed video streaming applications with the designed architecture without using any error correction codes to recover the data when it is affected by transmission errors. The proposed technique is more advantageous as it utilizes the designed architecture itself for correcting the fault and produces the uninterrupted transmission within a short delay so that the error correction in this case is 100%.

Keywords-- Video Transmission, Error Detection and Correction, Fault Coverage, Data Recovery.

I. INTRODUCTION

The advancement in hardware technology for multimedia applications has led to the improvement of storing data and transmitting it in a more flexible manner for all motion estimation video standards. The video streaming has led to the great advent in many applications like video conferencing for corporate industries and also capturing and sending the images in satellite applications which is useful for the scientists to analyze the alien objects. The objects which are streamed by the satellites are captured and are sent as images to the space research centers which are used to analyze the data. This information which is obtained by the satellites is in the form of electrical signals which are either two dimensional or 3-Dimensional in nature. These signals are digitalized to get a better quality by using compression technique [1] [2]. In digital compression technique, the frames of the original image are divided into blocks and the pixel values are taken which is coded into either lossless or lossy technique. The compression allows more data to be transmitted with the air as the medium in a given bandwidth. The technology used for compression has to provide a good quality which will be less affected by channel noise and provides a good quality digital picture. The technique used behind this

technology is motion compensation which uses filters to eradicate the noise and improve the signal quality. This technique is robust in nature and provides high quality which is mainly used for high definition applications. The digital video which is obtained after processing is transmitted in the form of packets of small size which improves the requirement in area allocation in a given bandwidth. In order to add security to the transmitted data, more authentication schemes namely watermarking is added which may be either visible or non-visible so that the data is not tampered by hackers [3]. As the media is a real time transmission more technical challenges leading to illegal manipulation of the data may take place at any time without the knowledge of the sender. In order to prevent this hashing has been introduced [4] which will provide an authentication so that the received information has the authenticated key which will provide the main source to decrypt the data when it is transmitted in the secure medium so that the data remains the same even after decoding it. The authenticated key is always generated at random by using cryptographic algorithms which is mainly generated by the request of the sender which is either of public key or private key which prevents the tampering of the data where the key is encoded along with the original image so that the receiver uses the same key or different key to decrypt the image as per the algorithm applied. The digital images which are used for video transmission usually undergo

single bit change when attacked by malicious functions. To overcome these problems, digital signature schemes have been proposed [5] which is dependent on the key which is used for encryption which becomes a difficult task for the attacker to break the pixel values where he can use it for his manipulation. The communication in a video system is done at a dedicated bandwidth by converting the analog signals to digital signals.

II. HIGH SPEED ARCHITECTURES FOR VIDEO PROCESSING

High Definition (HD) quality videos have now become a great demand in the consumer electronics industry because of its high picture quality. The transmissions of the HD videos are mainly done in wireless transmission mode as it is more convenient to design as more external cords are avoided and transmission is done by using radio waves or infrared. These signals are transmitted in the atmosphere with air as a medium. The key issue of the transmission mainly lies on data rate and distance. To attain high performance in transmission, reconfigurable architectures have been provided which lowers the computational activity in real time applications. The recent trend in the architecture design is to support single cycle operation when the inputs are fed at a maximum speed by taking into considerations the complexity of the design. This is mainly achieved by adding parallel structures to the system which will speed up the device by taking only 1 clock cycle for the processing. Apart from parallel structures, the design has to be taken into account to implement sequential design also which takes pipelining stages to reschedule the operations from the input of the design to the output and also the submodules implemented when the design has been implemented in algorithmic level which helps us to reduce the delay in the circuit when it is working at high frequency level [6]. The transmission systems used in video processing

mostly require hardware software co-simulation which is mainly done with interfaces which provides synchronization between the two sides which should be fast enough to support the video rates. To provide this synchronization, we use Field Programmable Gate Arrays (FPGAs) which acts as a hardware accelerator and also uses reconfigurable principle where the user can change the program whenever necessary according to his algorithm applied. The video signals mainly differ in timing issues depending upon the video length. In order to minimize the timing issues, we use configurable system by using frame buffers which reduces the timing issues and also supports all video standards [7]. These configurable systems when implemented on FPGAs results in better performance of the system as they can be redesigned by the user any time and can be transferred to the FPGA board to provide hardware accelerators for the hardware software co-simulation. This type of methodology uses algorithmic level software tools to design a communication protocols and Hardware Description Language (HDL) which is targeted on a FPGA to drive the software tool which are serially connected to each other to perform the DSP functions [8]. Apart from that the architecture designed for video application should be flexible to cover different applications ranging from low quality video to high quality video level and also should permit reprogrammability depending upon the platform where it is implemented. The implementation issue should not only support the acceleration but also to enable the high speed search engine to provide accessibility to the pixels to obtain high quality in digital streaming by using high quality programming. In DSP applications ranging from software level to hardware level. The biggest issue in video processing is fixing the image pattern [9] which has to be stored and processed by the FPGA in a limited bandwidth. The image is divided in the form of pixels which is also called as arrays and stored in memory

cells as shown in Fig.1 which is used for processing.

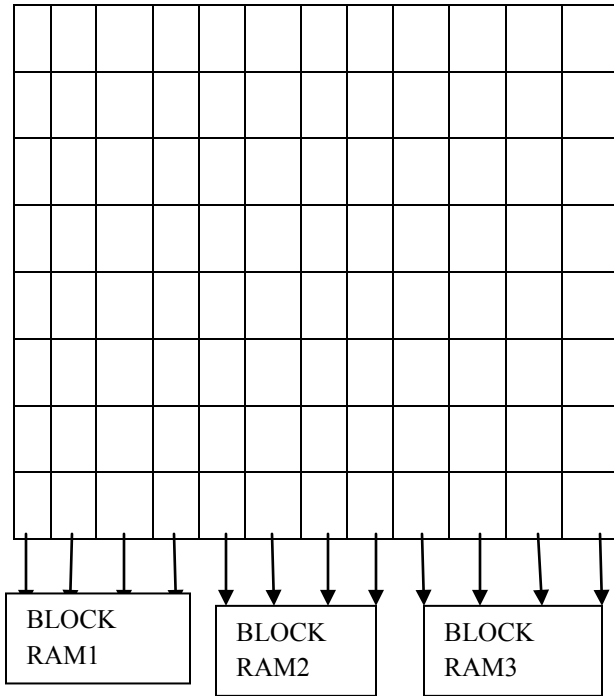


FIG.1 ARRAY STORAGE OF IMAGE PIXELS

Each array consists of memory address which is generated by the control signals and the data is obtained from the image pixel and stored at a particular location where the user can access the data quickly by using the search command. These instructions are dynamic which contribute the execution units in parallel computation technique where the instruction exploits parallelism which significantly accelerates media applications which helps in processing the support for area overhead and also reduces the critical path when the system operates at maximum clock speed [10]. The architecture can be either divided into 8X8 or 4X4 or 16X16 which is configured at the initial stage during the programming level by the user. The complexity of the high quality videos mainly depend on processing the intra prediction values which improves the coding efficiency for high picture quality. However the problem arises mainly due

to the complexity of the hardware elements which are used in the design which takes more clock cycles as the neighboring pixels are used as the reference values. To improve the architecture for intra prediction, parallel structure was proposed [11] which take 1 clock cycle to process 1 block. The computational complexity is further reduced by using combinational circuits used to compute the fast algorithm which consists of multiplexers (M) [11] which selects the boundary pixels (BP) and the coded pixels (CP) and computes the absolute value (ABS) as shown in Fig.2.

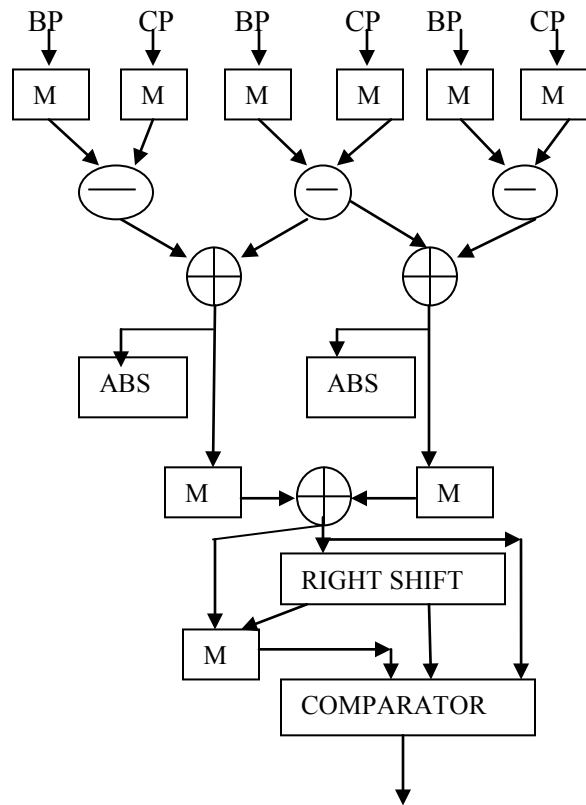


FIG.2. ABSOLUTE VALUE COMPUTATION

To further increase the speed of the design, we implement modified high speed adders with Binary To Excess-1 Logic (BEC) [12] with carry skip technique for 16 bits shown in FIG.3. The design can be made for low order and higher order by using logic shown in FIG.3

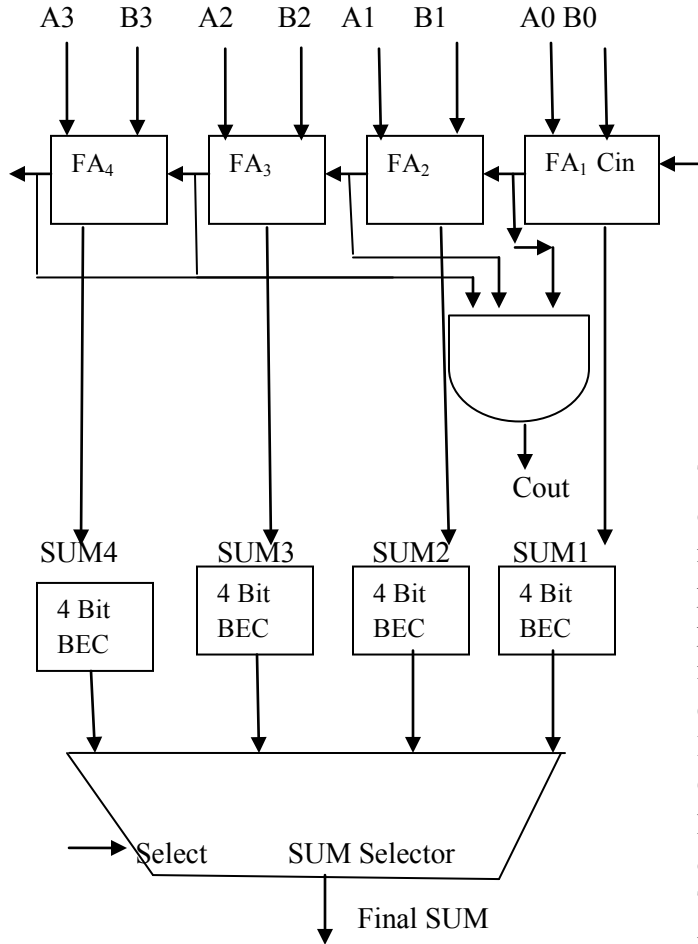


FIG.3. MODIFIED CARRY SKIP RIPPLE CARRY ADDER WITH BEC LOGIC

The design has been synthesized in Virtex-7 FPGA and the area, power and delay of 8 bit, 16 bit and 32 bit adder at a frequency of 100MHZ is shown in TABLE 1. The results show that the proposed design is more efficient than [12].

Adder Type	Area (No. Of Gates)	Power (mW)	Delay (ns)
8 bit	12	45	400
16 bit	38	57	789
32 bit	9	10	567

Table 1: AREA, POWER AND DELAY ANALYSIS

The number of intra pixels being utilized for the operation determines the improvement in the reduction of delay. Therefore high speed prediction unit has to be implemented processing element which will predict the intra mode pixels. The intra mode prediction unit consists of Fine Decision Unit (FD), Processing Element (PE) Unit and the Reusable Prediction Generator (PG) Unit [13]. The parallelism of the hardware mainly depends on the reconstruction of the data which determines the critical loop. The critical loop is reduced by introducing minimum pipeline structures for higher frequencies which reduces the execution time. Moreover the blocks are dependent on each other which cause interlacing technique thus reducing the throughput of the design by increasing the latency. This issue mainly occurs in the mode decision block (MD) and the reconstruction block (RB) when the data are sequentially processed. To overcome this problem, we introduce parallel processing of the four blocks simultaneously instead of two blocks in a Zigzag flow as shown in FIG.4

M	R _{3,N-3}	L _{3,N-3}	R _{3,N-2}	L _{3,N-2}	R _{3,N-1}	L _{3,N-1}	R _{3,N}	L _{3,N}
R		R _{3,N-3}	L _{3,N-3}	R _{3,N-2}	L _{3,N-2}	R _{3,N-1}	L _{3,N-1}	R _{3,N}
M			R _{3,N-3}	L _{3,N-3}	R _{3,N-2}	L _{3,N-2}	R _{3,N-1}	L _{3,N-1}
R			R _{3,N-3}	L _{3,N-3}	R _{3,N-2}	L _{3,N-2}	R _{3,N-1}	L _{3,N-1}

FIG.4 ZIG ZAG DIAGONAL SCANNING OF MD AND REC BLOCKS

With the implementation of the pipelined stage to 13, it takes 6 clock cycles to process the 96 pixel parallelism technique. With the minimization in clock cycles to process the overall pixels, the optimization of the data outputs to compute the inputs at high speed to produce the respective outputs which should be relevant to the inputs and also to process the data in parallel along with the pipelined technique. The design has been implemented in Xilinx Virtex-7 architecture as FPGAs provide high speed hardware acceleration with internally built parallel techniques for high speed data flow. The delay mainly depends on the number of pixels being operated at a time which also depends on the block size also. These are maintained at the buffer where the size depends on the image pixel size until the neighboring pixel is processed. This custom flow is mainly based on specific commands to capture the pixel and process it which should optimize automatically along with the target applications for high speed techniques. This technique will provide more flexibility for different computations to execute independently for all DSP applications. As the blocks works in parallel mode, the parallelism of the pipeline technique is reduced from 96 to 44 which is more advantageous than [13]. The computations required for processing the pixels are also

provided along with the pipelined architecture where the data is processed in parallel so that the clock cycle limits to 1. These hardware units processes the current state of the pixels and stores them in a memory where the address is provided by an external device .The throughput of the architecture mainly depends on the number of pixels processed per cycle at a given frequency of 110MHz. The processing speed of the pixels can be further increased by introducing a multiplying factor of a constant bit which will increase the number of pixels being processed at a given a given frequency which is given by the equation 1 for the pixels processed.

Throughput for Pixels Processed =

$$\frac{\text{Pixels}}{\text{Cycles}} \times \text{Constant Factor (1)}$$

Here the constant factor is defined by 2. Therefore the Pixels processed is given by

$$\frac{384}{256} \times 2 = 3 \text{ Pixels per Cycle. The maximum}$$

throughput of the design for the designed architecture to process the pixels is given by the equation 2

$$\frac{\text{Frequency}}{384} \times \text{Throughput of Pixels per Cycle}$$

(2) which equals $\frac{110\text{MHz}}{384} \times 3 = 859\text{k}$ per second at a given frequency which is more advantageous than [11].

III. REUSABLE ARCHITECTURE DESIGN

As the design utilizes flexible logic gates which can be reconfigured by the user. These devices are often exposed to critical environment which will influence the characteristic of the device which leads to unexpected operations and results in failure of the system. These failures are a

result from the radiation which causes extensive heat on the sensitive part of the device which causes degradation on the system where the components which are treated as black box where both the inputs and outputs are permanently failed during the normal operation and thus leading to unexpected outputs. These faults are modeled in digital terms by means of Struck At Faults which often induces errors in the computing devices. The modeling of these faults are done by detection, correction and the time required to manipulate the faulty block so that the circuit comes back to the normal position. The error detection and correction is basically based on the designed architecture or correction codes which will be useful for detecting multiple faults at a given time so that the efficiency of testing increases to a maximum without affecting the throughput of the design. The number of faults injected depends on the probability of total number of gates (n) to the total inputs (g) [14] which is given by the equation ${}^n C_g(3)$. Here we consider the fault in the adder circuit i.e. full adder whose total logic gates are 5 for 1 bit Full Adder (FA) as shown in FIG.5

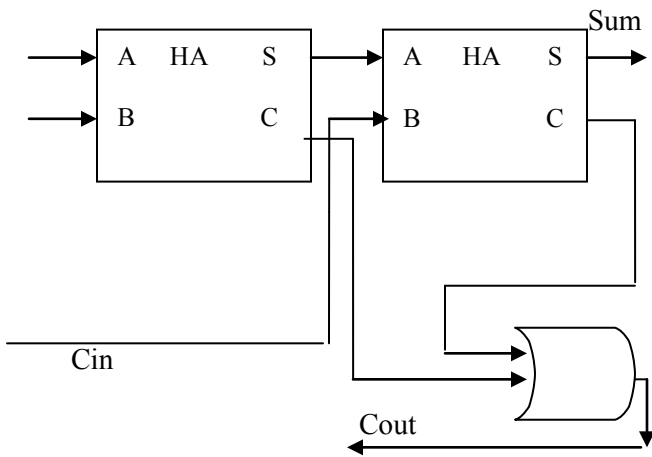


FIG.5 FULL ADDER USED FOR RIPPLE CARRY ADDER

Consider the failure of the 4 bit RCA as shown in FIG. 6 as shown in RED color.

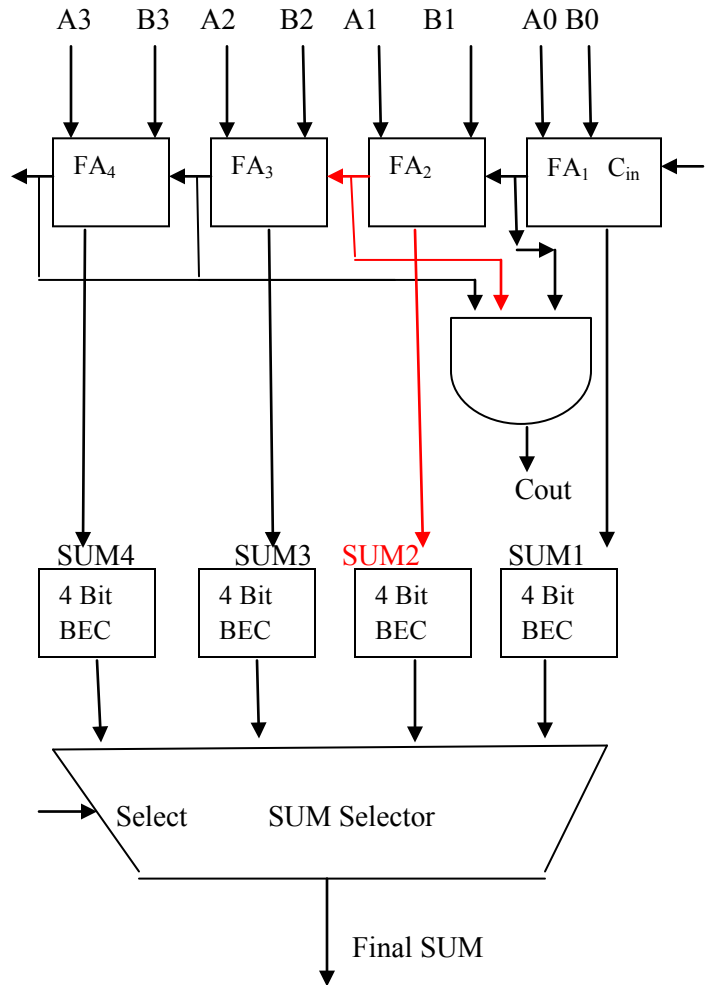


FIG.6. FAULTY ADDER BLOCK

For a 4 bit RCA, the number of possible combination of inputs are given by 16 and the number of gates are given by 8. So the total fault injection in the circuit is given by ${}^{16} C_8 = 12870$. The number of gate failures for the black box of 1 RCA is given by the probability of total gates used in the design to the total inputs and outputs which is given by ${}^7 C_6 = 7$ which equals to 1 black box. Here the number 6 stands for total inputs which is given by 3 and the total outputs which includes carry skip also which is given by 3 which equals 6. Consider the following JPEG color image which is to be transferred to the receiver from the satellite as shown in FIG.7.



FIG.7 JPEG IMAGE WHICH IS CAPTURED BY THE SATELLITE

Due to attacks, the receiver gets an error image which is not accurate as shown in FIG.8.



FIG.8. ERROR IMAGE WHICH IS OBTAINED BY THE RECEIVER

Once the exact image is not obtained by the receiver, he immediately sends an error message indicating the error signal to the satellite to recorrect the transmission. The error detection is mainly done with the help of checkpoints which are designed at the receiver end where he sends a signal to the transmitter indicating failure. The failure indication is indicated by the formula Failure = (Received Bits) XOR (Transmitted Bits) (4). If Failure = "0", an error has occurred in the transmission and so the receiver will send

an acknowledgment "0" to the receiver to recorrect the transmission. The checkpoints should be placed at proper intervals such that the execution of the program is not affected which in turn increase the latency of the system. The recovery time of the image should be independent of the fault location so that latency of the system is not increased which in turn does not create any interruption during transmission. The recovery time mainly depends on the faulty location which tries to extract the faulty image block as given by the equation $T_R = (nT + \Delta)$ [15] where "n" is the number of faulty frames, T is the Timeout Value and Δ is the data loading time. By our architecture developed, the recovery time gets reduced by $\frac{1}{2}$ as the system need get restarted to start the transmission which is more advantageous than [15]. The identification of the particular block which is failed is found by applying the test patterns. Once the block is identified, the correction method is one by using hardware reusable technique which takes the advantage of not applying error correction technique which is time consuming. In our method, the similar blocks which have the same functionality are routed together this applies the shortest path to reduce the delay as shown in FIG.9 where the faulty block generates the output "1" for all inputs which is applied. The other advantage of this process is that there is no hardware overhead as it does not increase the size of the overall architecture and does not utilize any algorithmic levels to correct the fault as in the previous method of algorithmic level fault tolerance technique [16]. The basic disadvantage of this technique is it uses more flip flops but in our design we need not construct any external hardware to recorrect the fault.

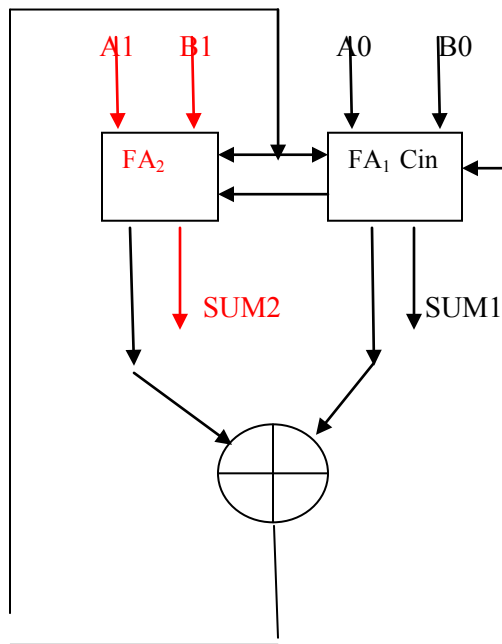


FIG.9 FAULT TOLERANT ARCHITECTURE

This architecture is routed internally between the modules having the same functionality which is used to detect the fault and send the command to the next module which is in the working state to take up the operation of the faulty module. Once the command is received, the inputs are transferred to the non faulty block so that the operation resumes without affecting the overall operation. As the patterns are applied parallel, the total clock cycle to perform the operation is “1” for the inputs and “1” for the output. During the fault condition, the total clock cycle in this case will be “3” as it has to detach from the non faulty block to the faulty block to process the inputs. The maximum throughput as shown in equation 2 gets reduced by a factor of 3 during the fault tolerant condition and hence the throughput is given by 856k per second for a given frequency. The fault detection is given by the probability analysis $F_{\text{detection}} = p^x \times {}^nC_x$ (4) where p is the probability of detecting the fault for the three inputs of the adder which is given by “1” and n is the number of inputs and x is the probability of success for all inputs. In this case with the help of check point insertion, the fault

detection will be 100% so that the error can be reported immediately for another transmission.

IV. RESULT AND CONCLUSION

We have presented a self tunable architecture which will auto correct the fault by using the principle of reusable technique within a short delay to start the process. This type of design is very useful for satellite applications which are used to send the captured images to the remote centre at times when it gets failed so that the receiver station need not send any correction technique to re-alter the process. Apart from that the system need not restart itself from the beginning to regenerate the process. This hardware fault tolerance presented in terms of auto correctability requires a short span which is to load the data once the fault is identified.

V. REFERENCE

- [1] David Taubman, MEMBER, IEEE, “High Performance Scalable Image Compression with EBCOT”, IEEE Transactions on Image Processing, 2000, vol 9, no.7, pp.1158-1170.
- [2] Vikrant Singh Thakur, Kavitha Thakur, “Design and Implementation of a highly efficient gray image compression codec using fuzzy based soft hybrid JPEG standard”, 2014 IEEE International Conference on Electronic Systems, Signal Processing and Computing Technologies. Nagpur (India), pp.484-489, 9-11 Jan. 2014.
- [3] Pradosh Bandyopadhyay, Soumik Das, Shauvik Paul, Atal Chaudhuri, Monalisa Banarjee, “A Dynamic Watermarking Scheme For Color Image Authentication”, 2009 IEEE International Conference On Advances in Recent Technologies in Communication and Computing. Kottayam, Kerala (India), pp. 314-318, 27-28 Oct. 2009.
- [4] Yao-Chung Lin, David Varodayan, Bernd Girod, “Image Authentication Based on Distributed source Coding”, 2007 IEEE

International Conference on Image Processing. San Antonio (TX). pp. 111-5 – 111-8, 16th Sept. 2007-19th Oct. 2007.

[5] Fawad Ahmed, M.Y.Siyal. “A Secure and Robust Hashing Scheme for Image Authentication”, IEEE 5th International Conference on Information, Communications and Signal Processing. Bangkok, pp. 705-709, Dec 6-9 Dec 2005.

[6] Rajesh S. Parthasarathy, Ramalingam Sridhar, “Double Pass Transistor Logic For High Performance Wave Pipelined Circuits”, 11th International Conference on VLSI Design. Chennai, pp. 495-500, 4-7 Jan 1998.

[7] Mustafa Dagtekin, Stephen C. Demarco, Rajeev Ramanath, Wesley E. Snyder, “A High Speed Video Processing and Display System. 2000 SPIE Medical Imaging Conference on Image Display and Visualization. San Diego (CA). 2000. pp.588-594, 12th Feb 2000.

[8] J.A. Kalomiros, J. Lygouras. Design and Evaluation of a Hardware / Software FPGA-based system for fast image processing. Elsevier Journals on Microprocessors and Microsystems, 2008. Vol.32, no.2, pp. 95-106.

[9] J. Dubois, M. Mattavelli, L. Pierrefeu, J. Miteran, “Configurable Motion Estimation Hardware Accelerator Module For the MPEG-4 Reference Hardware Description Platform”, 2005 IEEE International Conference On Image Processing. Genova. pp. 1040-1043, 11-14 Sept. 2005.

[10] Deependra Talla, Lizy.K.John, “Cost-Effective Hardware Acceleration of Multimedia Applications”, IEEE 2001 International Conference on Computer Design (ICCD, 2001). Austin (TX). pp. 415-424, 23-26 Sept 2001.

[11] Shih Chang Hsia, Ying – Chao Chou. VLSI Implementation of High Throughput Parallel H.264/ AVC baseline Intra Predictor. IET Journals on Circuits, Devices and Systems, 2013, vol.8, no.1, pp.10-18.

[12] Shivani Parmar, Kirat Pal Singh, “Design of High Speed Hybrid Carry Select Adder”, IEEE. 3rd International Conference on Advanced Computing Conference (IACC). Ghaziabad. pp. 1656- 1663, 22-23 Feb 2013.

[13] Gang He, Dajiang Zhou, Wei Fei, Zhixiang Chen, Jinjia Zhou, Satoshi Goto. High Performance H.264/AVC Intra Prediction Architecture for Ultra High Definition Video Applications. IEEE Transactions On Very Large Scale Integration (VLSI) Systems, 2014, vol.22, no.1, pp. 76-89.

[14] Mai C.R. De Vasconcelos, Denis T. Franco, LIRIDA A. De B. Naviner, Jean- Francois Naviner, “Reliability Analysis of Combinational Circuits Based on a Probabilistic Binomial Model”, 6th International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference. Montreal (QC). pp. 310-313, 22-25 June 2008.

[15] Sachin. P. Kamat, Senior Member IEEE. Fault- Tolerant Architecture for an MPEG-4 Based Video Decoder Driver. IEEE Embedded Systems Letters, 2012, vol.4, no.1. pp. 13-16.

[16] Adam Jacobs, Grzegorz Cieslewski, Alan D. George, “Overhead and Reliability Analysis Of Algorithm Based Fault Tolerance In FPGA Systems”, 22nd IEEE International Conference on Field Programmable Logic And Applications (FPL), Oslo (Norway), 2012, pp. 300-306, 29-31 Aug 2012.