International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 4, April- 2015

Efficient Image Transmission Using Space – Time Scheduling Strategy

, S.Md.Imran Ali[1] B.V.Ramana [2], G.Ayesha Anees[3], K.Madhu[4]
[1] *Asst Prof., Department of ECE, Brindavan Institute of Tech. & Science, Kurnool, A.P., India*
[2] *Asst Prof., Department of ECE, Brindavan Institute of Tech. & Science, Kurnool, A.P., India*
[3] *UG Student., ECE, Brindavan Institute of Tech. & Science, Kurnool, A.P., India*
[4] *UG Student., ECE, Brindavan Institute of Tech. & Science, Kurnool, A.P., India*

***Abstract* — The different optimization techniques are explored to improve the performance of the DCT. The latency of clock cycles of two dimensional DCT has been reduced by using pipelined and parallel processing of architecture. In the proposed system spatial scheduling strategy includes the ability to choose the distributed arithmetic (DA)-precision bit length, a hardware sharing architecture that reduces the hardware cost, and the proposed time scheduling strategy arranges different dimensional computations in that it can calculate first - dimensional and second - dimensional transformations simultaneously in single 1-D discrete cosine transform (DCT) core.**

The DA-precision bit length is chosen as 9 bits instead of the traditional 12 bits based on test image simulations. In addition, the proposed hardware sharing architecture employs a binary signed-digit DA architecture that enables the arithmetic resources to be shared during the four time slots. Multimedia applications, and in particular the en- coding and decoding of standard image and video formats, are usually a typical target for Systems- on-Chip (SoC).

***Index Terms*— Binary signed-digit (BSD), discrete cosine transform (DCT), distributed arithmetic (DA)-based, space-time scheduling (STS).**

I. INTRODUCTION

It has become very clear that all aspects of media are "going digital" from representation to transmission, from processing to retrieval, from studio to home. There have been significant advances in digital multimedia compression and communication algorithms, which make it possible to deliver high-quality video at relatively low bit rates in today's networks. The advancement in VLSI technologies has enabled sophisticated software to be implemented in a cost-effective manner. In this project we are going to transform the image along with the Image Transmission and the VLSI system using MATLAB and XILINX softwares.

The image transmission is used to transmit the image and it is compressed by the image compression. The image compression is used for reducing the amount of data required to represent an image. It is one of the most useful and commercially successful technologies in the field of digital image processing

The objective of image compression is to reduce irrelevance and redundancy of the image data in order to be able to store or transmit data in an efficient form. A chart showing the relative quality of various jpg settings and also compares saving a file as a jpg normally and using a "save for web" technique. Image compression may be lossy or lossless. Lossless compression is preferred for archival purposes and often for medical imaging, technical drawings, clip art, or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts.

Lossy methods are especially suitable for natural images such as photographs in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences may be called visually lossless. Image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the Internet or downloaded from Web pages. There are several different ways in which image files can be compressed.

For Internet use, the two most common compressed graphic image formats are the JPEG format and the GIF format. The JPEG method is more often used for photographs. While the GIF method is commonly used for line art and other images in which geometric shapes are relatively simple. Other techniques for image compression include the use of fractals and wavelets. These methods have not gained widespread acceptance for use on the Internet as of this writing. However, both methods offer promise because they offer higher

ISSN: 2278 – 909X

compression ratios than the JPEG or GIF methods for some types of images. Another new method that may in time replace the GIF format is the PNG format. A text file or program can be compressed without the introduction of errors, but only up to a certain extent. This is called lossless *compression*. Beyond this point, errors are introduced.

In image compression, a small loss in quality is usually not noticeable. There is no "critical point" up to which compression works perfectly, but beyond which it becomes impossible. When there is some tolerance for loss, the compression factor can be greater than it can when there is no loss tolerance. For this reason, gaphic images can be compressed more than text files or programs. Compressing an image is significantly different than compressing raw binary data. Of course, general purpose compression programs can be used to compress images, but the result is less than optimal. This is because images have certain statistical properties which can be exploited by encoders specifically designed for them. Also, some of the finer details in the image can be sacrificed for the sake of saving a little more bandwidth or storage space. This also means that lossy compression techniques can be used in this area.

Lossless compression involves with compressing data which, when decompressed, will be an exact replica of the original data. This is the case when binary data such as executables, documents etc. are compressed. They need to be exactly reproduced when decompressed. On the other hand, images (and music too) need not be reproduced 'exactly'. An approximation of the original image is enough for most purposes, as long as the error between the original and the compressed image is tolerable.

II. DISCRETE COSINE TRANSFORM

DISCRETE COSINE TRANSFORM (DCT) is a widely used transform engine for image and video compression applications [1]. In recent years, the development of visual media has been progressed towards high-resolution specifications, such as high definition television (HDTV). Consequently, a high-accuracy and high-throughput rate component is needed to meet future specifications. In addition, in order to reduce the manufacturing costs of the integrated circuit (IC), a low hardware cost design is also required. Therefore, a high performance video transform engine that utilizes high accuracy, a small area, and a high-throughput rate is desired for VLSI designs.

The 2-D DCT core design has often been implemented using either direct or indirect methods. The direct methods include fast algorithms that reduce the computation complexity by mapping and rotating the DCT transform into a complex number [3]. However, the structure of the DCT is not as regular as that of the fast Fourier transform (FFT). A regular 2-D DCT core using the direct method that derives the 2-D shifted FFT (SFFT) from the 2-D DCT algorithm format, and shares the hardware in FFT/IFFT/2-D DCT computation is implemented [6].

On the other hand, the 2-D DCT cores using the indirect method are implemented based on transpose memory and have the following two structures: 1) two 1-D DCT cores and one transpose memory (TMEM) and 2) a single 1-D DCT core and one TMEM. In the first structure, the 2-D DCT core has a high throughput rate, because the two 1-D DCT cores compute the transformation simultaneously [7]. In order to reduce the area overhead, a single 1-D DCT core is applied in the second structure,[2] thereby saving hardware costs. However, the additional input
buffers are needed in order to temporarily store the input data during the 2nd-D transformation.The 1-D DCT core can calculate the 1st-D and 2nd-D DCT computations simultaneously, and the throughput achieves 100 Mpels/s. However, multiplier-based processing element requires a large amount of circuit area. As a result, it is obvious that a tradeoff is required between the hardware cost and the speed. The balance point between the area required and the speed for 2-D DCT designs and present a multiplier-based pipeline fast 2-D DCT accelerator implemented using a field-programmable gate arrays (FPGA) platform. Additionally, several 2-D DCT designs are implemented based on FPGA for fast verification.

In this paper, an 8 X 8 2-D DCT core that consists of a single 1-D DCT core and one TMEM is proposed using a strategy known as space-time scheduling (STS). Due to the accuracy simulations in DA-based binary signed-digit (BSD) expression, 9-bit DA precision is chosen in order to meet the requirements of the peak-signal-to-noise-ratio (PSNR) outlined in previous works.

Furthermore, the proposed DCT core is designed based on a hardware sharing architecture with BSD DA-based computation so as to reduce the area cost. The arithmetics share the hardware resources during the four time slots in the DCT core design. A 100% hardware utilization is also achieved by using the proposed time scheduling strategy, and the 1st-D and 2nd-D DCT computations can be calculated at the same time.Therefore, a high performance transform engine with high accuracy, small area, and a high-throughput rate has been achieved in this research.

III. PROPOSED METHOD AND ITS DESCRIPTION

To save hardware costs, the proposed 2-D DCT core, as shown in Fig.3.1 , is implemented using a single 1-D DCT core and one TMEM. The 1-D DCT core includes an MBF2, a Pre-Reorder module, a PEE, a PEO, a Post-Reorder module, and one TMEM. The TMEM is

implemented using 64-word 12-bit dual-port registers and has a latency of 52 cycles. Based on the time scheduling strategy described in Time Scheduling Strategy, a hardware utilization of 100% can be achieved.

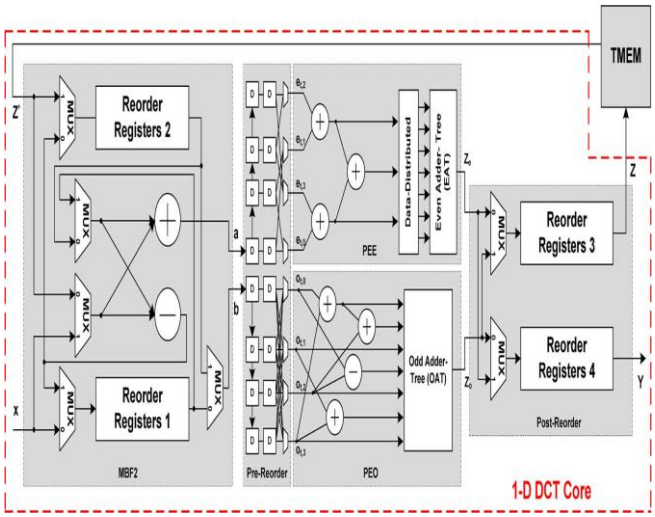


Fig 1 : Proposed 2-D DCT architecture

a) MODIFIED BUTTERFLY MODULE

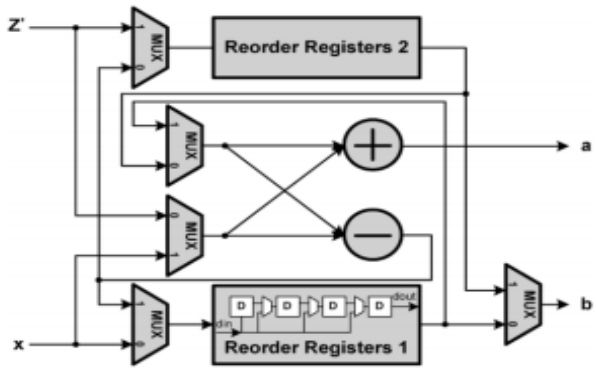


Fig 2 : MBF2 Architecture

Modified Butterfly Module is easily implemented using a two-input butterfly module called BF2. In general, the BF2 has a hardware utilization rate in the adder and subtracter of 50%. In order to enable the hardware resources to be shared additional multiplexers and Reorder Registers are added to the proposed MBF2 module. The Reorder Registers consist of four word registers that use the control signals to select the input data and use enable signals to output or hold the data. Similar to BF2 the operation of the proposed MBF2 has an eight-clock cycle period. In the first four cycles the 1st-D input data (x0, x1, x2, and x3) shift into Reorder Registers 1 and the 2nd-D input Z data execute the operation of addition and subtraction. In the next four cycles the operations of 1st-D and 2nd-D will be changed.

b) PROCESSING ELEMENT

DA-based computation the even part and odd part transformations can be implemented using PEE and PEO respectively. The even part transformation can be expanded for the DA-based computation formats and can share the hardware resources at the bit level. The coefficient vector has two combinations of [C4C4C4C4] and [C2C6C6C2] for (Z0, Z4) and (Z2, Z6) transform outputs respectively.

The EAT sums the different weighted values in tree-like adders to complete the transform output Ze. Similarly the odd part transformation it can be implemented in a DA based format using four adders and one odd part adder-tree (OAT). The EAT and OAT can be implemented using the error-compensated adder tree to improve the computation accuracy. Moreover there are three pipeline stages (two in both the EAT and the OAT) in each PEE and PEO module that enable high speed computation to be achieved. Inputs enter into the processing element from the upper left.

The first step is for each of these inputs to be multiplied by their respective weighing factor (w(n)). Then these modified inputs are fed into the summing function which usually just sums these products. Yet many different types of operations can be selected. These operations could produce a number of different values which are then propagated forward values such as the average the largest the smallest the OR –ed values the AND –ed values etc. Furthermore most commercial development products allow software engineers to create their own summing functions via routines coded in a higher level language (C is commonly supported).

Sometimes the summing function is further complicated by the addition of an activation function which enables the summing function to operate in a time sensitive way. Either way the output of the summing function is then sent into a transfer function. This function then turns this number into a real output via some algorithm. It is this algorithm that takes the input and turns it into a zero or a one a minus one or a one or some other number.

The transfer functions that are commonly supported are sigmoid sine hyperbolic tangent etc. This transfer function also can scale the output or control its value via thresholds. The result of the transfer function is usually the direct output of the processing element. In the process element there are primary and secondary colours which are separated by the even and odd functions. There is a distributed arithmetic which distributes the data and given input to the even adder tree and to the odd adder tree.

ISSN: 2278 – 909X

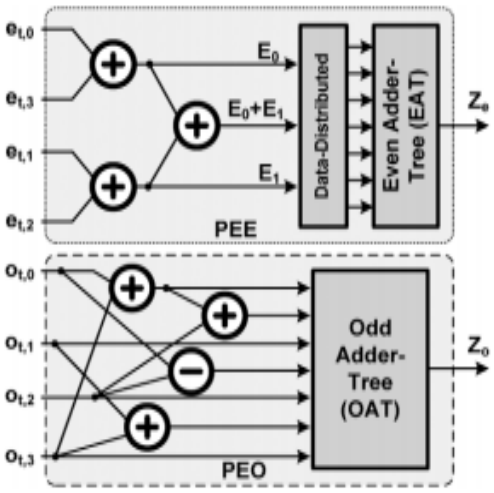


Fig 3 : PEE and PEO Architecture

c) POST – REORDER

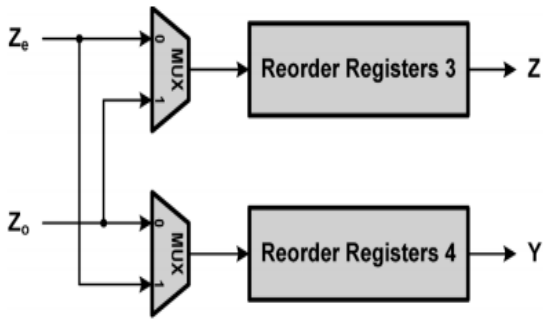


Fig 4 : Post-Reorder

The data sequence after the PEE and PEO must be merged and re-permuted in the Post-Reorder module.The order of the original 8-point data sequence from the PEE and the PEO is {Z0, Z2, Z4, Z6, Z1, Z3, Z5, and Z7}.After the Post-Reorder module permutes the data order the data sequence is ordered as {Z0, Z2, Z4, Z6, Z1, Z3, Z5, Z7 }. Two multiplexers select the data that is fed into the different Reorder Registers in order to permute the output order. Z is the transform output for the 1st-D DCT that will input into the TMEM. In addition the 2nd-D DCT transform output is completed after the permutation by Reorder Registers 4.

TIME SCHEDULING STRATEGY

As a result of the time scheduling strategy, the 1st-D and 2nd-D transforms are able to be computed simultaneously, which increases hardware utilization be 100%. The timing flow chart for the proposed strategy is illustrated in Fig. .

1) 1st-D Data Computation:

In the first four cycles, the first four-point input data shift into the Reorder Register 1. During cycles 5–8, the MBF2 performs additions and subtractions using the first eight - point input data. In the 9th cycle, the even-part of Pre-Reorder module obtains and reorders the input data output to the PEE. During cycles 10–13, the is calculated in the PEE module. Based on the latency of the three pipeline stages in the PEE, is completed during cycles 13–16. Also, this is performed in the PEO module during cycles 14–17 and is finished in cycles 17–20. Therefore, after the 16th cycle, the Post-Reorder module permutes the 1st-D transform results and inputs them into the TMEM.

2) 2nd-D Data Computation:

From the 17th cycle, the 1st-D transform data is input into the TMEM. Due to the latency of the TMEM 52 cycles , the 2nd-D computation data transposed by the TMEM is sent to the input of Reorder Registers 2 in MBF2 at the 69th cycle. Then, the adder and subtracter are operated during cycles 73–76 to compute the first 2nd-D 8-point data sequences, and the first 2nd-D data is permuted by Pre-Reorder module during cycles 77–80. The PEE and PEO calculate the first 2nd-D data during cycles 78–81 and 82–85, respectively. The Post-Reorder module finishes the 2nd-D transform results from the 84th cycle. In addition, the 2-D DCT transform output data is obtained at the end of 84th cycle.

3) Hardware Utilization:

In Fig. 6, the adder and subtracter in MBF2 is at 50% utilization during cycles 1–68. However, after the 68th cycle, the MBF2 achieves a 100% hardware utilization as a result of the 2nd-D data being input. At the same time, the utilization in the PEE and PEO also reaches 100% from 74th and 78th cycles, respectively. The Post-Reorder module does not work between cycles 1–12, but after the 16th cycle the 1st-D transform output are completed using Reorder Registers 3 in the Post-Reorder. Similarly, the 2nd-D transform result is finished using Reorder Registers 4 in the Post-Reorder from the 84th cycle. At the end of 84th cycle, the 1st-D and 2nd-D transform outputs and are obtained simultaneously. Hence, the hardware utilization increases to 100% after the 84th cycle, and the latency of the proposed 8*8 2-D DCT core is 84 clock cycles. Based on the proposed time scheduling strategy, a 100% hardware utilization is achieved, and computation of the data sequence is straight forward.

In summary, following an analysis of the accuracy analysis of the coefficients, the DA-precision bit length was chosen as a 9-bit BSD expression. In this way,

the hardware cost is reduced by using the BSD 9-bit DA-precision instead of the 12-bit used in previous works. Using the proposed hardware sharing strategy, the DCT computation shares the hardware resources during the four time slots, thereby reducing the area cost. In addition, a 100% hardware utilization can be achieved based on the proposed time scheduling strategy, effectively achieving a high throughput- rate design. Therefore, a high-accuracy, small-area, and high-throughput-rate 2-D DCT core has been obtained by applying the proposed STS strategy.

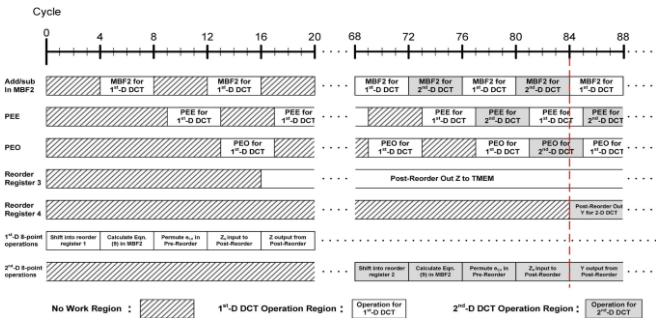


Fig 5 : Time Scheduling Strategy

IV. SIMULATION RESULTS

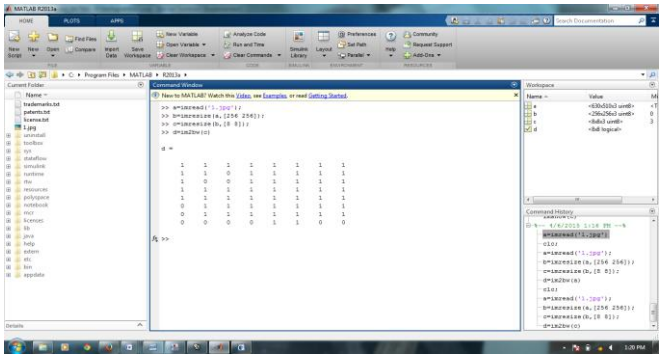


Fig 6 : Output of an image in binary

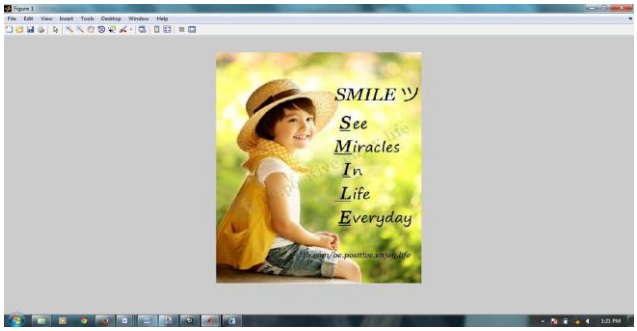


Fig 7 : Original image taken as input in MATLAB

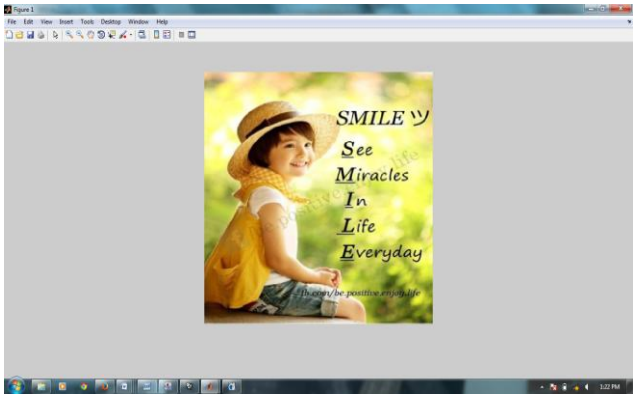


Fig 8 : The image is compressed with 256X256 pixel

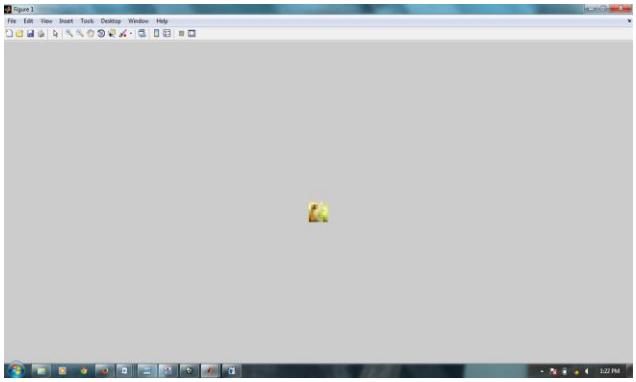
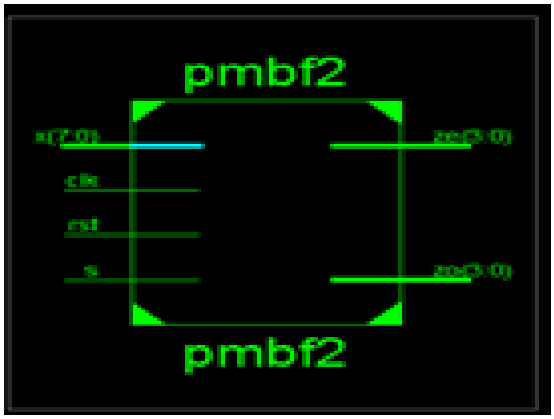


Fig 9 : The image is compressed with 8X8 pixel



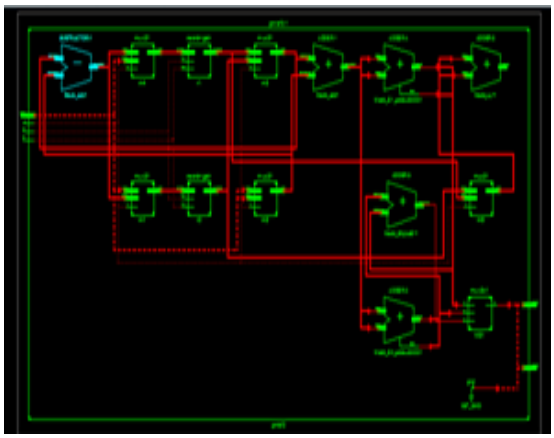


Fig 10 : RTL schematic in Xilinx

ISSN: 2278 – 909X

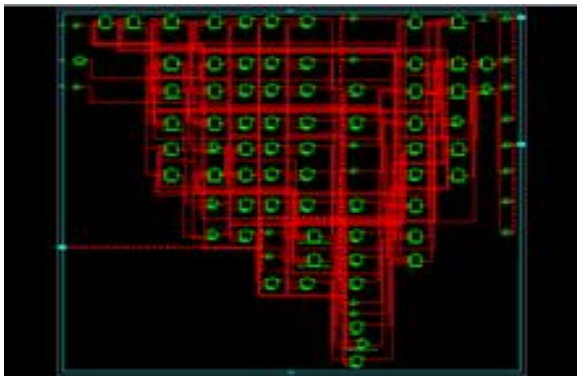


Fig 11 : Technology Schematic

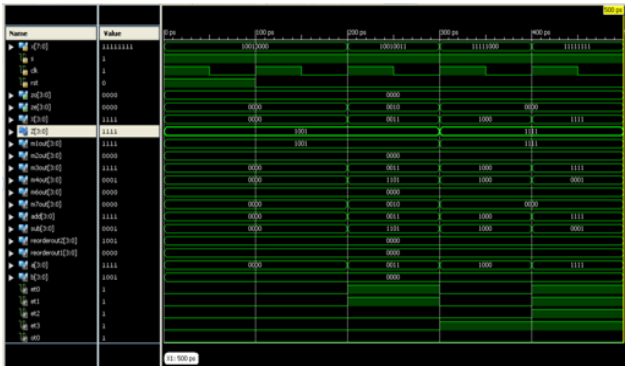


Fig 12 : Wave form in Xilinx

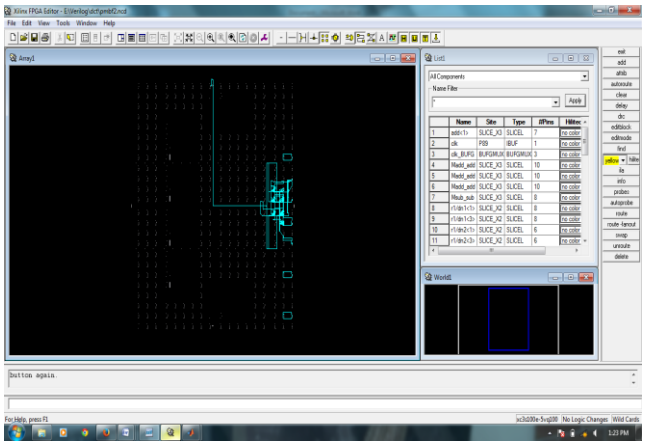


Fig 13 : Placed in FPGA

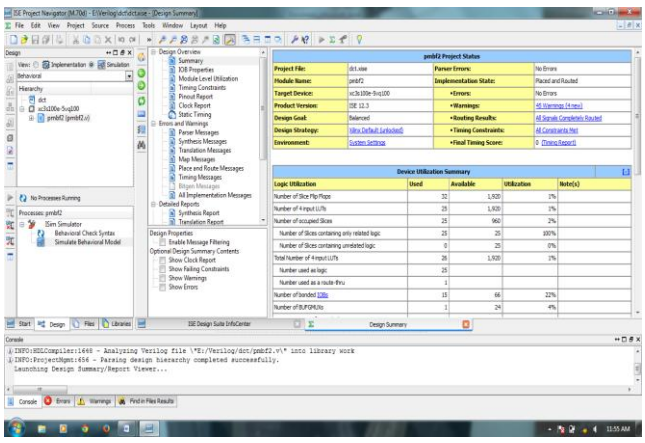


Fig 14 : Design Summary

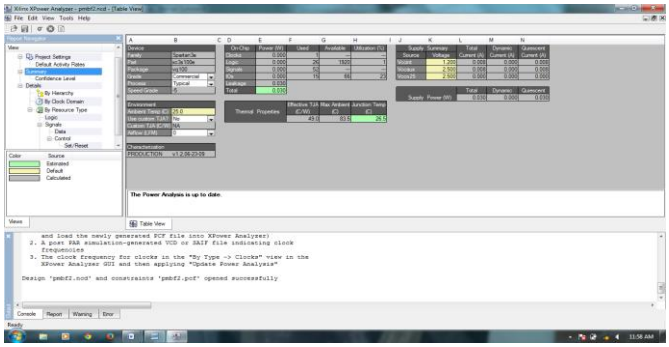


Fig 15 : Reduction of power

V. CONCLUSION AND FUTURE SCOPE

The proposed spatial scheduling strategy includes the capacity to select the distributed arithmetic (DA) – precision bit length, a hardware sharing architecture that reduces the hardware cost, and the proposed time scheduling strategy arranges different dimensional computations in that it can calculate 1-D and 2-D transformations at the same time in single 1-D discrete cosine transform (DCT) core. The structure of Re-order Register is customized and designed and its output is obtained. The DA-precision bit length is preferred as 9 bits as an alternative of the usual 12 bits based on test image simulations. The reorder register is implemented in the 2-D DCT architecture and the area is reduced.

In future we can enhance the image, as the resolution of the image decreases the quality of the image is reduced due to compression. Hence we can go for various transformations and this can be applicable for video transform.There is another method called Discrete Wavelet Transform (DWT) which can be preferable in enhancing the image/video transform.

REFERENCES

[1] A. M. Shams, A. Chidanandan, W. Pan, and M. A. Bayoumi, "NEDA: A low-power high performance DCT architecture," IEEE Trans. Signal Process., vol. 54, no. 3, pp. 955–964, Mar. 2006.

[2] A. Madisetti and A. N. Willson, "A 100 MHz 2-D 8 X 8 DCT/IDCT processor for HDTV applications," IEEE Trans. Circuits Syst. Video Technol., vol. 5, no. 2, pp. 158–165, Apr. 1995.

[3] A. Tumeo, M. Monchiero, G. Palermo, F. Ferrandi, and D. Sciuto, "A pipelined fast 2-D DCT accelerator for FPGA-based SOCs," in Proc. IEEE Comput. Soc. Annu. Symp. VLSI., 2007, pp. 331– 336.

[4] C. C. Sun, P. Donner, and J. Gotze, "Low complexity multi-purpose IP core for quantized discrete cosine and integer transform," in Proc. IEEE Int. Symp. Circuits Syst.,2009, pp. 3014–3017.

[5] C. Peng, X. Cao, D. Yu, and X. Zhang, "A 250 MHz optimized distributed architecture of 2D 8 x 8 DCT," in Proc. Int. Conf. ASIC, 2007, pp. 189–192.

[6] C. T. Lin, Y. C. Yu, and L. D. Van, "Costeffective triple-mode reconfigurable pipeline FFT/IFFT/2-D DCT processor," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 16, no. 8, pp. 1058–1071, Aug. 2008.
[7] C. Y. Huang, L. F. Chen, and Y. K. Lai, "A highspeed 2-D transform architecture with unique kernel for multi-standard video applications," in Proc. IEEE Int. Symp. Circuits Syst., 2008, pp. 21–24.39
[8] E. Feig and S.Winograd, "Fast algorithms for the discrete cosine transform," IEEE Trans. Signal Process., vol. 40, no. 9, pp. 2174–2193, Sep. 1992.
[9] H. C. Hsu, K. B. Lee, N. Y. Chang, and T. S. Chang, "Architecture design of shape adaptive discrete cosine transform and its inverse for MPEG-4 video coding," IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 3, pp. 375–386, Mar. 2008.
[10] J. I. Guo, R. C. Ju, and J. W. Chen, "An efficient 2-D DCT/IDCT core design using cyclic convolution and adder-based realization," IEEE Trans. Circuits Syst. Video Technol., vol. 14, no. 4, pp. 416–428, Apr. 2004.

AUTHOR'S PROFILE

Mr. S.Md. Imran Ali has persued his B.Tech from SAFA college of Engg, & Tec and M.Tech from KTMC. Presently he is working as Asst. Prof in Dept of ECE of Brindavan Institute of Technology & Sciences, Kurnool. His research areas of interest are VLSI for signal and image processing, wireless Communications, Antenna Design, Analog & Digital Communications, Digital Integrated Circuit Design Kurnool.

Mr. B.V.RAMANA has persued his B.Tech from St JOHNS college of Engg, & Tech.Yemmiganoor and M.Tech from Sree Vidyanikethan Engineering College, Tirupathi Presently he is working as Asst.Prof in Dept of ECE of Brindavan Institute of Technology & Sciences, Kurnool. His research areas of interest are Electronic Circuit Analysis & Design, Limaer Integrated Circuit Design, Fault detection and testing of digital circuits, Fault Model Analysis.

Ms.G.Ayesha Anees is persuing her B. TECH in Brindavan Institute Of Technology & Science. She is from Dept of ECE from BITS, Kurnool. Her areas of interest in VLSI designing.

Mr.K.Madhu is persuing his B.TECH in Brindavan Institute Of Technology & Science. He is from Dept of ECE from BITS, Kurnool. His areas of interest in VLSI designing.

ISSN: 2278 – 909X