



Input Based Dynamic Reconfiguration for Low Power Image Processing and Secure Transmission

Jadi Raju

M.Tech., CMR Institute of
Technology, Kandlakoya (V),
Medchal (M), Hyderabad,
Telangana, India

MD. Shabazkhan

Associate Professor, ECE
Department, CMR Institute of
Technology, Kandlakoya (V),
Medchal (M), Hyderabad,
Telangana, India

G. Laxmi Narayana

Assistant Professor, ECE
Department, CMR Institute of
Technology, Kandlakoya (V),
Medchal (M), Hyderabad,
Telangana, India

ABSTRACT

The image and video processing algorithms are very compute intensive and with increase in resolution, the width of the compute elements like adders, etc. increase and this increase the power consumption of the device by several times. Approximate computing can reduce the power consumption as careful approximation does not affect the output quality of the image and video. Fixed levels approximation yield inconsistent quality output for different images and videos. In this project, we propose a dynamic approximation based image processing circuit. We implement an input based dynamically approximate reconfigurable adders and sub-tractors who can adjust their level of approximation dynamically by looking at the input thrown to them and thus, can trade-off between quality and power saving. We implement the code in Verilog HDL and verify the power by using the power estimator in Xilinx ISE tool. The simulation will be demonstrated in Modelsim software.

Keywords: Approximate circuits, zig-zag coding, low power design, quality configurable

1. Introduction

In today's world where electronics are becoming cheaper due to the advancement in the semiconductor areas and also due to the research and development in other areas of science and technology like optics, sensors, etc. we are getting high quality image and video capture devices. A typical 10 mega pixel photo would occupy over 40MB of space which is difficult to store or transmit and the problem is more severe when

it comes to video where a 1 second video contains at least 25 frames and therefore, would occupy 1000MB i.e. approximately 1GB and hence, there is a necessity to compress the images and videos for storage and transmission. Image compression may be lossy or lossless. Preferred for archival purposes and often for medical imaging Lossless compression is used for accurate results, technical drawings, clip art, or comics. Lossy compression methods, especially when used at low bit rates, introduce compression artifacts. The 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. Lossy compression that produces negligible differences may be called visually lossless.

2. Decompression

In decompression, the steps of compression are performed in the reverse order i.e. Inverse run length coding, Inverse ZigZag coding, inverse quantization and Inverse DCT. Due to lossy compressions, the recovered image is not exactly equal to the original image.

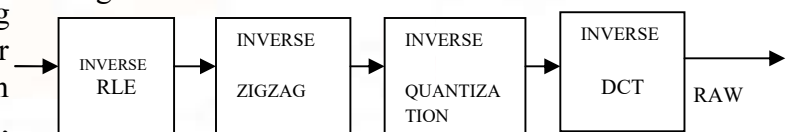


Figure 1: Decompression block diagram

Above block diagram explained below.

Inverse Run length

The compressed frame is passed through the inverse run length encoding process. The inverse run length encoder reads the marker which says the quantity of repetition of its succeeding character and outputs the value quantity number of times thus giving the same output of the zigzag encoder in the compression scheme

Inverse zigzag

Inverse Zigzag process re arranges the values of the matrix received in the order before they were scrambled by the ZigZag transformation in the compression scheme. The output of the Inverse zigzag should look exactly like the output of the Quantization step in the quantization process.

Inverse Quantization

Inverse quantization multiplies each of the matrix variable by the corresponding quality matrix variable. This is not a matrix multiplication but, only an element by element multiplication. Since quantization is a lossy transformation, inverse quantization does not yield the exact result as the output of the DCT process. The reduction in quality is the price we pay for the compression.

Inverse DCT

Inverse DCT is the exact inverse transformation of the DCT transformation. As a result of the inverse DCT process the matrix changes from the frequency domain to the amplitude domain and we can see the pixel values now. The decompressed image is not exactly like the original image as we have lost some information due to quantization and some information due to the rounding off in DCT and IDCT.

$$Original = \begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}$$

$$Decompressed = \begin{bmatrix} 149 & 134 & 119 & 116 & 121 & 126 & 127 & 128 \\ 204 & 168 & 140 & 144 & 155 & 150 & 135 & 125 \\ 253 & 195 & 155 & 166 & 183 & 165 & 131 & 111 \\ 245 & 185 & 148 & 166 & 184 & 160 & 124 & 107 \\ 188 & 149 & 132 & 155 & 172 & 159 & 141 & 136 \\ 132 & 123 & 125 & 143 & 160 & 166 & 168 & 171 \\ 109 & 119 & 126 & 128 & 139 & 158 & 168 & 166 \\ 111 & 127 & 127 & 114 & 118 & 141 & 147 & 135 \end{bmatrix}$$

The above matrices show that there are slight differences between original image and decompressed image but, the above compression reduces 70% space or bandwidth.

3. Approximate Computing

Approximate computing has made its way into image and signal processing big time in the current generation where the algorithms used are compute extensive and slower machines are not preferred for automation these days. In this project we replace the ripple carry adders with their approximate version Reconfigurable adder/ subtractor blocks (RABs). Reconfigurable Adder/Subtractor Blocks Dynamic variation of the DA can be done when each often adder/subtractor blocks is equipped with one or more of its approximate copies and it is able to switch between them as per requirement. This reconfigurable architecture can include any approximate version of the adders/subtractors. As a reference, Gupta et al. proposed six different kinds of approximate circuits for adders. However, it also needs to be ensured that the additional area overheads required for constructing the reconfigurable approximate circuits are minimal with sufficiently large power savings. As examples, we have chosen the two most naive methods presented, namely, truncation and approximation 5, for approximating the adder/subtractor blocks. The latter one can also be conceptualized as an enhanced version of truncation as it just relays the two 1-bit inputs, one as Sum and the other as Carry Out (Choice 2). In case A, B, and Cin are the 1-bit inputs to the full adder (FA), then the outputs are Sum = B and Cout = A. The resultant truth-table [10] shows that the outputs are correct for more than half of all input combinations, thus proving to be a better approximation mode than truncation. The proposed scheme replaces each FA cell of the adders/subtractors with a dual-mode FA (DMFA) cell (Figure 3.1) in which each FA cell can operate either in fully accurate or in some

approximation mode depending on the state other control signal APP. A logic high value of the APP signal denotes that the DMFA is operating in the approximate mode.

We term these adders/subtractors as RABs. It is significant to note that the FA cell is power-gated when operating in the approximate mode.

Our experiments have shown a negligible difference in the power consumption of DMFA when operated in either of the two approximation modes. Hence, without any loss of generality, approximation 5 was chosen for its higher probability of giving the correct output result than truncation, which invariably outputs 0 irrespective of the input.

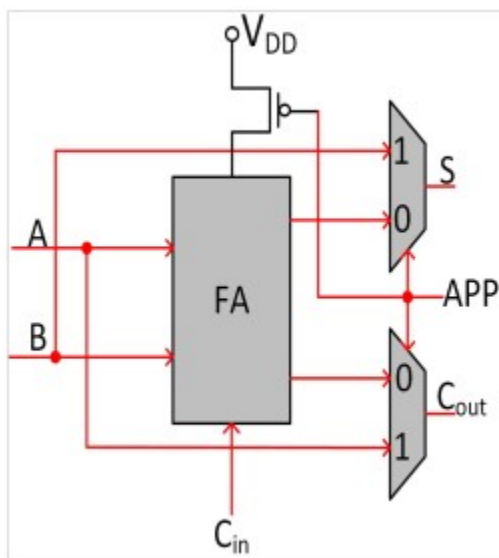


Figure 2: bit DMFA

Figure 2 shows the logic block diagram of the DMFA cell, which replaces the constituent FA cells of an 8-bit RCA,

This undermines the primary objective as most of the power savings that we get from approximating the bits are lost. Instead, the two-mode decoder and the 2:1 multiplexers have negligible overhead and also provide sufficient command over the approximation degree. 1) DMFA Overhead: The power gating transistor and the multiplexers of the DMFA are designed to incur the least possible overhead.

Our experiments show that switching power of the CMOS transistors contributes toward most of the total power consumption of the FA and DMFA blocks. Table I presents the power consumption of FA and DMFA for different modes obtained by exhaustive simulation in Synopsys NanoSim.

It shows that the power increases by 0.21 μ W when we operate DMFA in accurate mode as compared with the

original FA block. This difference in power can be attributed mainly to the increase in load capacitance of the FA block due to the addition of the input capacitance of the interfaced multiplexers. A small portion of the total power is contributed by the additional switching of the multiplexers. Table I also shows that the power consumed during DMFA approximate mode is almost negligible when compared with the accurate mode, which is due to the power gating of the FA block by the pMOS transistor, as shown in Figure 3.1. Reduction in the input switching activity of the multiplexers is also a secondary cause for this small amount of power.

4. RESULTS

4.1 SIMULATION and SYNTHESIS RESULTS

In this section, we show the simulation results of various blocks like DCT, quantization, zigzag and run length encoding.

We are starting with the given matrix

$$M = \begin{bmatrix} 26 & -5 & -5 & -5 & -5 & -5 & -5 & 8 \\ 64 & 52 & 8 & 26 & 26 & 26 & 8 & -18 \\ 126 & 70 & 26 & 26 & 52 & 26 & -5 & -5 \\ 111 & 52 & 8 & 52 & 52 & 38 & -5 & -5 \\ 52 & 26 & 8 & 39 & 38 & 21 & 8 & 8 \\ 0 & 8 & -5 & 8 & 26 & 52 & 70 & 26 \\ -5 & -23 & -18 & 21 & 8 & 8 & 52 & 38 \\ -18 & 8 & -5 & -5 & -5 & 8 & 26 & 8 \end{bmatrix}$$

Inverse RLE Simulation Output:

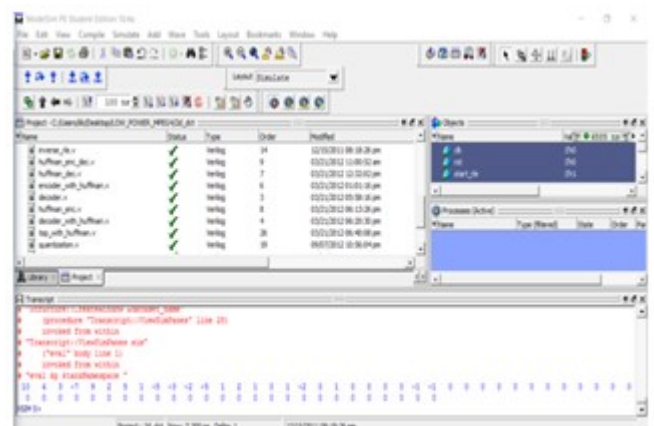


Figure 3: Inverse RLE Output

The Figure 3 shows the Inverse RLE Output compressed frame is passed through the inverse run length encoding process. The inverse run length encoder reads the marker which says the quantity of repetition of its succeeding character and outputs the value quantity number of times thus giving the same output of the zigzag encoder in the compression scheme.

RTL Schematic of Inverse RLE

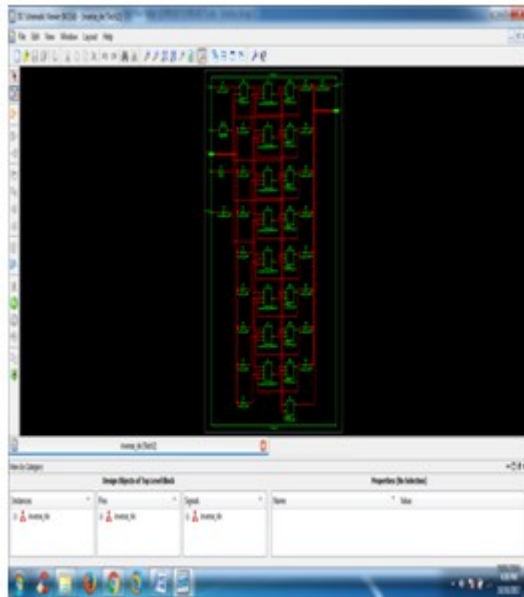


Figure 4: RTL schematic of inverse RLE

Above figure 4 shows 4 RTL schematic of inverse RLE. In this schematic all registers and transistors are in a logic is employed and resultant output generated.

TECH schematic of inverse RLE

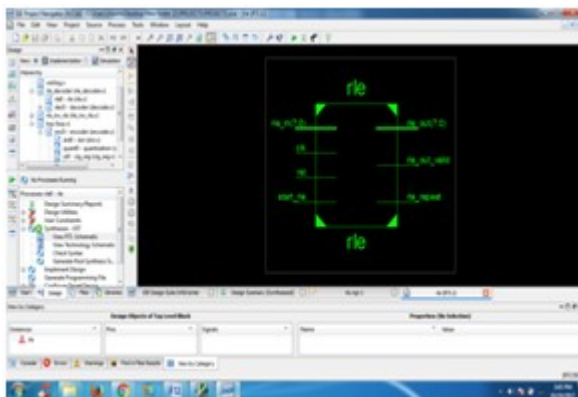


Figure 5: TECH schematic of inverse RLE.

Above figure 5 shows TECH schematic of inverse RLE.

Inverse Zig Zag Simulation Output:

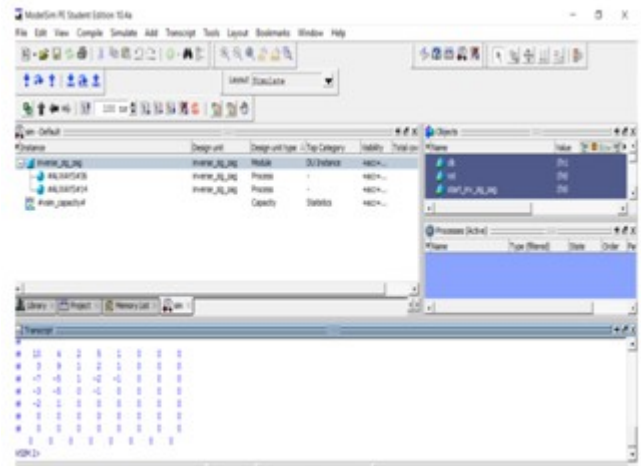


Figure 6: Inverse Zig Zag Output

Above figure-6 shows the Inverse Zigzag process re arranges the values of the matrix received in the order before they were scrambled by the ZigZag transformation in the compression scheme. The output of the Inverse zigzag should look exactly like the output of the Quantization step in the quantization process.

Synthesis summary of Inverse RLE

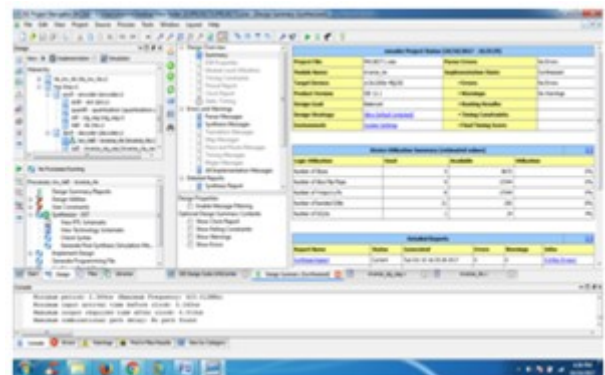


Figure 7 Synthesis summary of Inverse RLE

Above figure 7 shows Synthesis summary of Inverse RLE in this all the values of the image pixel values in this Synthesis summary are arranged.

RTL schematic of Inverse of RLE

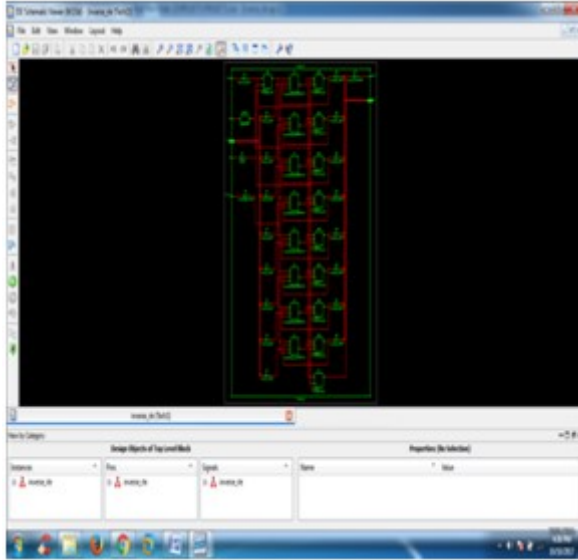


Figure 8 RTL schematic of Inverse of RLE

Figure 8 shows the RTL schematic of Inverse of RLE. It consist of register and transistors it arranging number of image are expanding and showing into next step

TECH Schematic of Inverse RLE

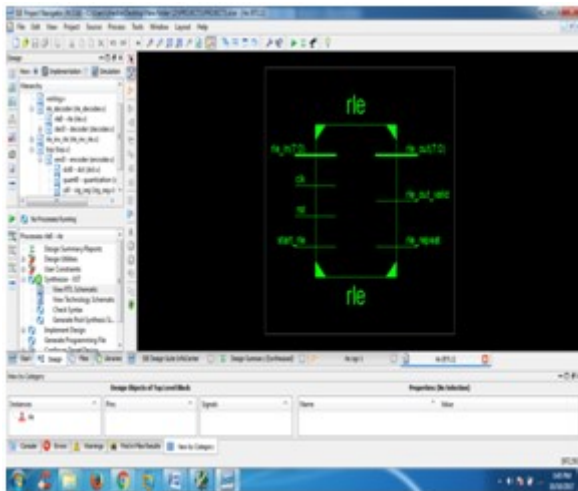


Figure 9 TECH Schematic of Inverse RLE

Above figure 9 shows TECH schematic of inverse RLE.

Synthesis summary of Inverse ZIG ZAG

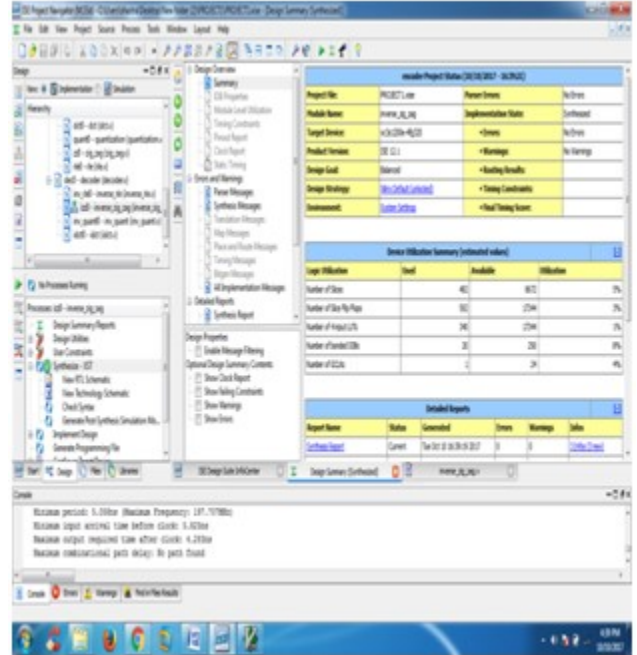


Figure 10: Synthesis summary of Inverse ZIG ZAG

Above figure 10 shows Synthesis summary of Inverse ZIG ZAG in this all the values of the image pixel values in this Synthesis summary.

RTL schematic of Inverse ZIG ZAG

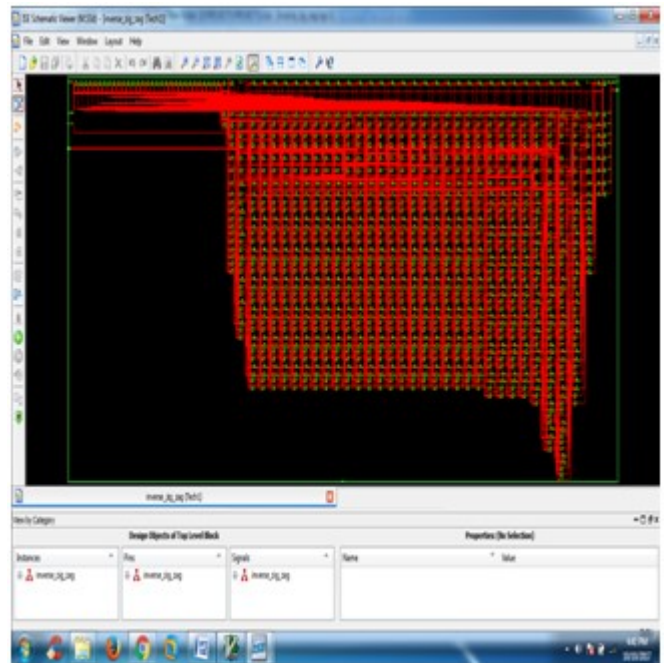


Figure 11 RTL schematic of Inverse ZIG ZAG.

Above figure 11 shows the schematic of Inverse ZIG ZAG in this all registers and transistor are arranged to arrange the all image pixel values are re-arranged in proper.

TECH schematic of Inverse QUANT

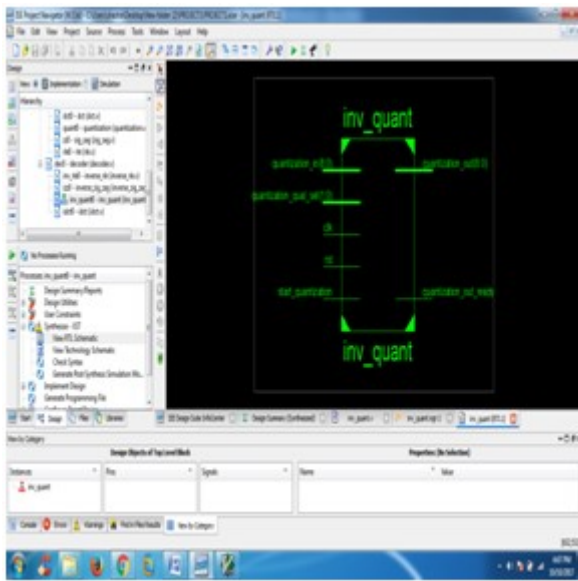


Figure 16: TECH schematic of Inverse QUANT

Above Figure 16 shows Tech Schematic of Inverse QUANT.

Synthesis summary of Inverse DCT

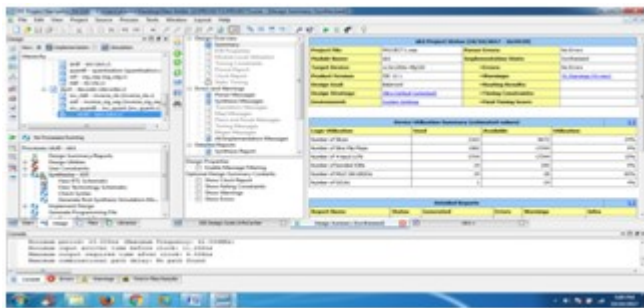


Figure 17 Synthesis summary of Inverse DCT

Above figure 17 shows Synthesis summary of Inverse DCT in this all the values of the image pixel values in matrix Synthesis summary mathematically in this.

TECH schematic of Inverse DCT

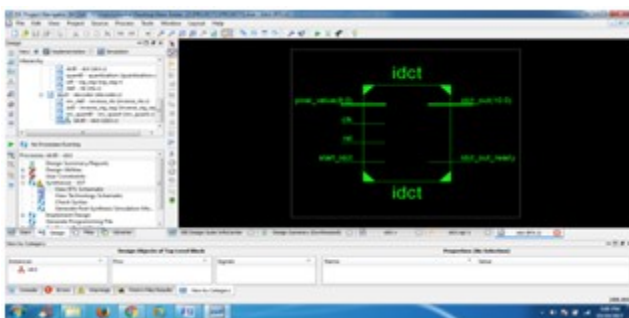


Figure 18 RTL schematic of Inverse DCT

Above Figure 18 shows Tech Schematic of Inverse DCT.

Synthesis summary of DECODER

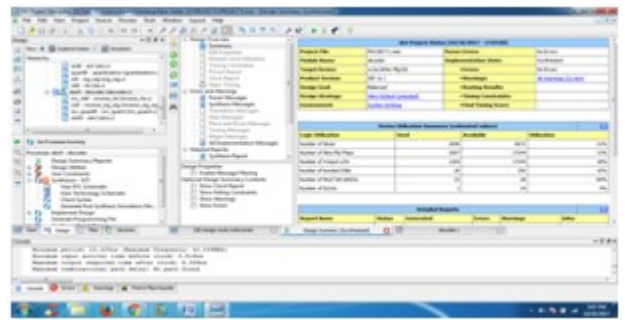


Figure 19: Synthesis summary of DECODER

Above figure 19 shows the Synthesis summary of DECODER and which values are taken in the decoder is shown in this summary.

TECH schematic of Decoder

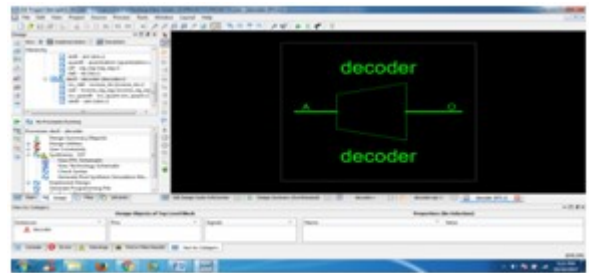


Figure 20 TECH schematic of Decoder.

Above Figure 20 shows Tech Schematic of Decoder.

Synthesis summary of Top Module

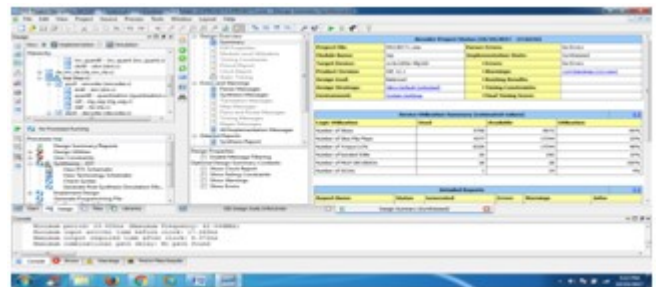


Figure 21: Synthesis summary of Top Module

Above figure 21 shows Synthesis summary of Top Module in this decoder values are shows it consist the total summary of the decoder.

TECH schematic of Top module

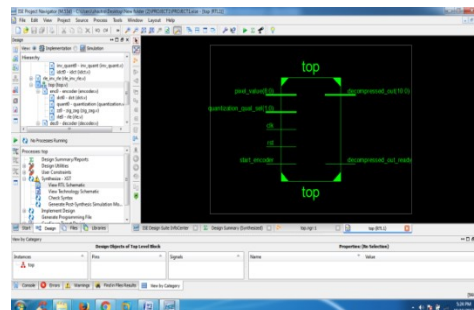


Figure 22 RTL schematic of Top module

Above Figure 22 shows the RTL schematic of Top module of the decoder and it showing number of pins in the decoder.

5. SUMMARY**5.1 APPLICATIONS**

DCT is used mostly in Image compression and image encoding. High speed approximate image compression can be used in

- High speed camera circuits like professional photography, high performance mobile cameras, etc.
- High speed image processing such as facial recognition circuits and electronic microscopes.
- High speed image processing like analyzing images taken by satellites, space telescopes, etc.

5.2 CONCLUSION

- We have successfully implemented the approximate adder circuit
- We have applied the approximate adder in the DCT circuit and we have demonstrated the functioning of DCT with approximate adders.
- We have implemented the entire image encoding flow – DCT, Quantization, ZigZag encoding and Run length encoding.
- We have demonstrated the variations in compression ratios with variations in quality levels.

5.3 FUTURE SCOPE

- The circuit can be further improved by converting the 2D DCT into approximate DCT which can be implemented without any multipliers and thus, the circuit complexity decreases greatly.

There are approximate multipliers which have been proposed in some recent works which will further help in increasing the speed.

REFERENCES

1. M. Elgamel, A. M. Shams, and M. A. Bayoumi, "A comparative analysis for low power motion estimation VLSI architectures," in Proc. IEEE Workshop Signal Process. Syst. (SiPS), Oct. 2000,
2. F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," Proc. IEEE, vol. 83, no. 6, pp. 858–876, Jun. 1995.
3. I. S. Chong and A. Ortega, "Dynamic voltage scaling algorithms for power constrained motion estimation," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), vol. 2, Apr. 2007, pp. II-101–II-104.
4. I. S. Chong and A. Ortega, "Power efficient motion estimation using multiple imprecise metric computations," in Proc. IEEE Int. Conf. Multimedia Expo, Jul. 2007, pp. 2046–2049.
5. D. Mohapatra, G. Karakonstantis, and K. Roy, "Significance driven computation: A voltage-scalable, variation-aware, quality-tuning motion estimator," in Proc. 14th ACM/IEEE Int. Symp. Low Power Electron. Design (ISLPED), 2009, pp. 195–200.
6. J. George, B. Marr, B. E. S. Akgul, and K. V. Palem, "Probabilistic arithmetic and energy efficient embedded signal processing," in Proc. Int. Conf. Compil., Archit., Synth. Embedded Syst. (CASES), 2006, pp. 158–168.
7. D. Shin and S. K. Gupta, "A re-design technique for datapath modules in error tolerant applications," in Proc. 17th Asian Test Symp. (ATS), 2008, pp. 431–437.
8. S. Venkataramani, A. Sabne, V. Kozhikkottu, K. Roy, and A. Raghunathan, "SALSA: Systematic logic synthesis of approximate circuits," in Proc. 49th Annu. Design Autom. Conf. (DAC), Jun. 2012, pp. 796–801.
9. V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "IMPACT: IMPrecise adders for low-power approximate computing," in Proc. 17th IEEE/ACM Int. Symp. Low-Power Electron. Design (ISLPED), Aug. 2011, pp. 409–414.
10. V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low power digital signal processing using approximate adders," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.
11. V. G. Moshnyaga, K. Inoue, and M. Fukagawa, "Reducing energy consumption of video memory by bit-width compression," in Proc. Int. Symp. Low Power Electron. Design (ISLPED), 2002, pp. 142–147.
12. Z. He and M. L. Liou, "Reducing hardware complexity of motion estimation algorithms using

truncated pixels,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), vol. 4. Jun. 1997, pp. 2809–2812.

13. Z.-L. He, C.-Y. Tsui, K.-K. Chan, and M. L. Liou, “Low-power VLSI design for motion estimation using adaptive pixel truncation,” IEEE Trans. Circuits Syst. Video Technol., vol. 10, no. 5, pp. 669–678, Aug. 2000.
14. A. Raha, H. Jayakumar, and V. Raghunathan, “A power efficient videoencoder using reconfigurable approximate arithmetic units,” in Proc. 27th Int. Conf. VLSI Design, 13th Int. Conf. Embedded Syst., Jan. 2014, pp. 324–329.
15. P. M. Kuhn, Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation, 1st ed. Norwell, MA, USA: Kluwer, 1999.
16. K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, “Study of subjective and objective quality assessment of video,” IEEE Trans. Image Process., vol. 19, no. 6, pp. 1427–1441, Jun. 2010.
17. S. Winkler, “Video quality measurement standards—Current status and trends,” in Proc. 7th Int. Conf. Inf., Commun., Signal Process. (ICICS), Dec. 2009, pp. 1–5.
18. Xiph.org. (2013). Xiph Video Test Media. [Online]. Available: <http://media.xiph.org/video/derf/>
19. Wikipedia. (2013). Carry-Select Adder—From Wikipedia, the Free Encyclopedia. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Carry-select_adder&oldid=572157817, accessed Jan. 30, 2014.

Authors Profile:



Jadi.Raju received his bachelor’s degree in 2015 in electronics and communication engineering from JJ institute of technology, India which is affiliated with JNTU Hyderabad, India. His areas of interest include VLSI design. He is pursuing his M-Tech in VLSI SYSTEM DESIGN from CMR Institute of Technology.

Mr. Mohammad Shahbaz Khan M.TECH (PhD), Associate Professor, ECE Department, CMRIT, He has 10 years of teaching experience. He has worked as Associate Professor in various Engineering colleges under JNTUH such as Amina institute of Technology, he coordinated & organized many workshops, FDP and contributed 20 papers to various conference & Journals



Mr. G. Lakshminarayana M.Tech (Ph.D), Assistant professor, ECE Department, CMRIT. He has completed M.Tech with specialization in Digital Systems and Computer Electronics (D.S.C.E) from JNTUHCE (Autonomous), Hyderabad in 2010 and A.M.I.E (E.C.E) from IEI, Kolkata, India in winter 2004. He has 12 Years of academic experience at Various Engineering Colleges. He has immense research and academic experience in Digital Systems. His Research Areas of Interest are VLSI, Digital Systems and Embedded Systems.