Journal of VLSI circuits and systems
RESEARCH ARTICLE
Journal of VLSI Circuits and Systems, ISSN: 2582-1458 Vol. 6, No. 2, 2024 (pp. 40-52)
WWW.VLSIJOURNAL.COM

Low Power System on Chip Implementation of Adaptive Intra Frame and Hierarchical Motion Estimation in H.265

T M Praneeth Naidu[1*], P. Chandra Sekhar[2], Pradeep Kumar Boya[3]
[1]Research Scholar, Department of ECE, University College of Engineering, Osmania University, Hyderabad, 500007.
[2]Professor, Department of ECE, University College of Engineering, Osmania University, Hyderabad, 500007.
[3]Sr. Research Engineer, Incline Inventions Pvt. Ltd, Hyderabad, India.

KEYWORDS:
H.265,
System on Chip (SoC),
FPGA,
Adaptive Intra-Frame Prediction, Hierarchical Motion Estimation,
Video Compression,
Real-Time Encoding,
Low Power

ARTICLE HISTORY:
Received 03.04.2024
Accepted 19.05.2024
Published 11.06.2024

DOI:
https://doi.org/10.31838/jvcs/06.02.05

Abstract

Over the years, video compression standards have evolved and helped distribute multimedia content with greater ease. H.265 or High Efficiency Video Coding (HEVC) is one of the best lossy video codecs released as successor to H264 which offers better compression. This analysis is focused on tackling the computation needs of H.265 encoding through System on Chip (SoC) based implementations. The SoC platform has both ARM and FPGA architectures to handle floating point and fixed-point operations with parallelism respectively. The intended methodology is implemented using adaptive intra-frame prediction and hierarchical motion estimation suitable for SoC implementation to bridge the current gap. The innovation of this approach is that the parallel processing capabilities built into SoCs are used to improve intra-frame prediction and motion estimation efficiency. This proposed method involves the adaptive mechanisms for intra-frame prediction as well as a hierarchical structure of motion estimation to balance computational load and resource consumption. In conventional intra prediction, all the 35 modes are implemented, and the best mode is selected based on the Sum of Absolute Difference (SAD) value. In SoC this process can be implemented in parallel by using non-blocking modules for all the 35 modes. In order to improve the efficiency further, edge detection and content analysis modules are added to analyse the blocks before the modes are selected. This make sures that only required modes are executed based on the content that is present in the block. The traditional motion estimation module is improved by adding a hierarchical framework to select the best matching block more effectively. To solve the power consumption problems, Intelligent Clock Gating (ICG) is applied in order to switch off idle modules and hence reduce dynamic power consumption drastically. This low-power optimization results in extended operation times for portable devices while minimizing thermal footprint. The proposed methodology successfully improved the overall efficiency of the H.265 framework.

Author e-mail: praneethtm@gmail.com, sekharp@osmania.ac.in, pradeep.boge@gmail.com

How to cite this article: Naidu PTM, Sekhar CP, Boya PK. Low Power System on Chip Implementation of Adaptive Intra Frame and Hierarchical Motion Estimation in H.265, Journal of VLSI Circuits and System Vol. 6, No. 2, 2024 (pp. 40-52).

1. INTRODUCTION

The development of video compression standards has dramatically altered the performance level and quality of multimedia material distribution. H.265 (known as High Efficiency Video Coding - HEVC) is also among the better-known technologies developed using these standards and it offers a higher compression capability compared to its predecessors.[1] H.265 achieves the same by employing more advanced techniques like improved intra-frame prediction, effective motion estimation and suitable entropy coding among others.[2] These improvements have enabled much lower bit rates to be used whilst maintaining visual quality, accommodating for the increased requirement of HD video streaming and storage.

H.265 has a lot going for it but because of the complex nature of encoding, there are many challenges that lie upstream as well. Motion Estimation: Motion estimation

is a key block in inter-frame prediction, but employing an exhaustive search for finding optimal motion vectors makes it computationally intensive.[3] Just like that, intra-frame prediction investigates each mode in greater depth to figure out which one is best for a given block. These operations, however, even though needed for high-compression efficiency also introduce computational load and encoding overhead that may be prohibitive in real-time or resource-constrained applications.[4]

As the amount of high-resolution video materials increases, and on-demand processing in real-time as a principle is demanded for encoding H.265 effectively, However, the native limitations in software-based tools often prevent them from providing enough performance to manage the computing requirements imposed by H.265 encoding.[5] This limitation underscores the need for hardware acceleration solutions to be available as soon as possible, and particularly SoC-based ones that bring parallel processing capabilities along with adaptive designs catered well towards the specific algorithmic functionalities.[6]

However, H.265 Encoding on SoCs bemires with problems of its own However, novel design methodologies are required for H.265 to achieve high performance due to the intrinsic complexity of its motion estimate and intra-frame prediction algorithms.[7] There are several important aspects - high throughput needs to be preserved, power consumption comes into play and an efficient use of resources is necessary. It needs lot of design and optimization effort to balance between compression efficiency as well hardware complexity.[8]

Adaptive intra-frame and hierarchical motion estimation for H.265 that is especially designed for SoC implementation is the suggested method's approach to overcoming these obstacles. This method improves motion estimation and intra-frame prediction efficiency by taking use of SoCs parallel processing capabilities. To maximize the computational burden and resource consumption, the suggested technique uses an adaptive mechanism for intra-frame prediction and a hierarchical structure for motion estimation. Thus, a solid answer to the difficulties involved in H.265 encoding is provided, enabling real-time encoding speed while preserving excellent compression efficiency.

In FPGA implementations of H.265 encoding, low power consumption is especially important for portable and battery-operated devices. Low power implementations can also extend operational time and reduce the energy footprint that video processing tasks need. Optimizing power with high performance is crucial to meet the demands of modern, low-power electronics. While designing the low power strategies, the balance between performance and power efficiency needs to be maintained. There are several reasons why low power consumption is important in order to implement H.265 encoding on SoC and FPGAs. Many video encoding applications, e.g., portable devices, mobile platforms and remote surveillance systems, are often run by battery power. It allows the devices to last longer without being re-charged and thus without needing a new battery harder. Low-power SoC based H.265 implementations can significantly lower bandwidth requirements when transferring video footage over networks. This is very important in Surveillance applications because the bandwidth may be limited or costly. Low power H.265 implementations can encode and decode video content in real-time under lower scenarios which is important for processing of high-resolution imaging streams, such images transmitted by surveillance systems.

2. Literature

Grzegorz Pastuszak et al[9] demonstrated the design of the entropy coder, which can handle a lot more binary symbols each clock cycle than earlier efforts. The design makes use of the binary arithmetic coder (BAC) multisymbol implementation that was previously established. Quick implementations of context modeling, probability model (PM) updating, and binarization are designed to balance large throughputs of the BAC. The primary increase in the symbol rate is a result of the processing path being divided into several parallel ones.

Sushanta Gogoi et al[10] suggested a hardware architecture and quick hybrid search pattern algorithm for encoding UHD videos. Compared to the Test Zone Search (TZS) method by default in the HM reference software, the suggested Integer ME (IME) technique needs an average of 11.19% less encoding time without compromising PSNR degradation and bit rate increase.

Sushanta Gogoi et al[11] provided a hardware implementation of an IME algorithm with minimal complexity. The authors have shown the suggested approach on two distinct pattern structures (PS) with 38 search points. In comparison to test zone search (TZS) in HM 16.8, it achieves bjontegaard bitrate (BD-BR) of 0.5% drop and 0.077% increase for two PSs. The encoding time is reduced by 8.725% and 9.072%, respectively. Every HEVC partition is supported by the suggested architecture.

Chi-Ting Ni et al[12] provided a high-efficiency, hardware-friendly H.265/HEVC acceleration method to get around this complication for networks of visual sensors.

The suggested technique speeds up intra-prediction for intra-frame encoding by utilizing texture direction and complexity to avoid doing unnecessary processing in the CU partition. One popular video compression standard is high-efficiency video coding (HEVC/H.265). In comparison to H.264/AVC, which can compress visual data with a high compression ratio but resulting in significant computational complexity, HEVC reduces the bit rate by around 50% at the same video quality.

Abdulaziz Alarifi et al[13] proposed a unique hybrid cryptosystem for safe streaming of compressed HEVC streams that combines Mandelbrot sets, the Arnold chaotic map, and DNA (Deoxyribonucleic Acid) sequences. First, the H.265/HEVC codec is used to encode the high-resolution movies in order to obtain effective compression performance. Next, the proposed Arnold chaotic map ciphering procedure is applied separately to the three compressed HEVC frame channels (Y, U, and V). The principal encrypted frames from the earlier chaotic ciphering procedure are then used to establish the DNA encoding sequences. Subsequently, a conditional shift mechanism based on a modified Mandelbrot set is used to create confusion characteristics on the Y, U, and V channels of the ciphered frames.

Cuong Pham-Quoc et al[14] offered an effective hardware/software codesign strategy to use FPGA-based IoT edge computing platforms to speed up the video encoder process. In order to boost system performance, the design flow makes use of the high-level synthesis process' ability to swiftly and effectively construct FPGA-based hardware accelerator cores. As a case study, the authors validate the suggested design flow and assess the accelerator performance in relation to general-purpose processors using the H.264 encoder.

Fritjof Steinert et al[15] provided a unique H.264/Advanced Video Codec (AVC) based video codec that enables the coding of certain areas of an image with nearly constant picture quality to enable a dependable neural network-based classification, while the rest image will be coded with a constant bit rate. The authors have coupled this functionality with the ability to operate with the lowest possible latency attributes, which is typically also necessary in instances involving remote control applications.

Yufan Lu et al[16] suggested a novel adaptable hardware accelerator framework that would allow an FPGA-based edge computing platform to handle different DL algorithms in an adaptive manner. This framework enables reconfiguration at run-time to match the demands of unique application specifications and operating conditions, hence increasing the power and computational efficiency of both DNN model/software and hardware.

3. Proposed Method

H.265./High Efficiency Video Coding (HEVC) is a video compression standard, designed to substantially improve coding efficiency compared to its predecessor, the AVC/H.264. The workflow of H.265 video compression consists of several steps in a pipeline to compress the very high quality. The blocks in the workflow of H.265 are:

1. Preprocessing

The preprocessing consists of steps like noise reduction and frame resizing. This process helps in effectively compressing the data using the HEVC algorithm.

2. Partitioning into Coding Tree Units (CTUs)

Instead of the fixed-size macroblocks that were used in H.264, H.265 uses a flexible block structure (large blocks are sub-divided into smaller ones), called Coding Tree Units (CTUs). The geometry of this rectangular CTU can be adjusted, with common sizes between 16x16 and 64x64 pixels, to better suit different ranges of content. In turn, each CTU can be divided into Coding Units (CUs), Prediction Units (PUs) and Transform Unit (TUs) for more fine-grained compression.

3. Intra Prediction

The goal of intra prediction is to reduce the spatial redundancy within a frame. Intra prediction involves the use of neighbouring pixel values within the same frame to predict the current pixel values. H.265 supports 33 intra prediction modes, compared to just 9 in H.264, which leads to much more accurate predictions and hence better compression.

4. Inter Prediction

Inter prediction eliminates the temporal redundancy in adjacent frames by predicting the current frame's content using one or more reference frames. Real-time video coding is mainly based on the motion estimation and compensation techniques. The H.265 codec improves the compression efficiency using a large number of precise motion vectors and multiple reference frames

5. Transform and Quantization

After the prediction block, a Discrete Cosine Transform (DCT) or Discrete Sine Transform (DST) is applied on residuals which are differences between predicted and

true pixel values. These transforms transform spatial time domain data into frequency domain data, and help transform the signal pixel data in just a few coefficients. These coefficients are quantised to eliminate redundancy by eliminating less important coefficients.

6. Entropy Coding

The quantized coefficients as well the other compression parameters are coded using an entropy coding methodology. H.265 introduces two entropy coding methods: Context-Adaptive Binary Arithmetic Coding (CABAC) and Context-Adaptive Variable-Length Coding (CAVLC). While CABAC is more computation expensive compared to CAVLC, it provides better compression efficiency and thus more frequently employed in H.265.

7. Deblocking Filter and Sample Adaptive Offset (SAO)

H.265 uses deblocking filter to decrease the artefacts in order to improve visual quality; The reconstructed image is also further processed using the Sample Adaptive Offset (SAO) in order to correct distortions such as ringing that are introduced during quantization.

8. Rate Control

Dynamic Rate control is important for matching the video bit rate after compression to a target constraint (e.g., 1 or 3 Mbps file size per minute of average content, network bandwidth). It utilizes advanced rate control algorithms, dynamic quantization parameters and encoding specifics to deliver consistent video quality with minimal bandwidth consumption.

9. Encoding and Packaging

The workflow concludes by encoding the compressed video data into a bitstream format for storage or transmission. H.265 bitstreams are usually delivered in containers like MP4, MKV or MPEG-2 transport Stream hence bundling up the video, audio and maybe additional data into one file-infrastructure.

Low power design is necessary to avoid heat generation, which is very important to improve reliability and reduce thermal induced failures. High temperature is a major concern for electronic components, which leads to performance degradation and longevity. Low power systems are becoming essentially important for environmental sustainability. Reducing the power footprint of electronic systems helps save energy, which in turn supports ongoing global initiatives aimed at reducing carbon emissions and cultivating environmentally friendly technologies.

The proposed model uses multiple strategies to minimize power consumption during the H.265 encoding procedure. Utilizing techniques that facilitate power saving enable the model to improve performance while also providing sustainability and energy-efficient video encoding, which is suitable for low-power applications as well as mobile devices.

3.1 Adaptive Intra Prediction

The prediction of the current block is done by referencing pixels from left and top side of the block. These are the reference pixel blocks which comes from boundary of already encoded and reconstructed neighbor block. The reference pixels along this angle are used to compute the prediction for each pixel in that block. If a prediction angle does not exactly align with the reference pixels, interpolation is used to calculate what the values of these predicted held will be. HEVC uses linear interpolation between the reference pixels to decide the value of the predicted pixel. Suppose the prediction point is between two reference pixels. In that case, the pixel value is obtained using a weighted average of the two reference pixels. After the prediction is made, the residual between the actual pixel values and the predicted values is calculated and encoded. The coded residual information, and the mode information thus obtained from all modes is sent to the decoder. Figure 1 shows the intra modes of H265.

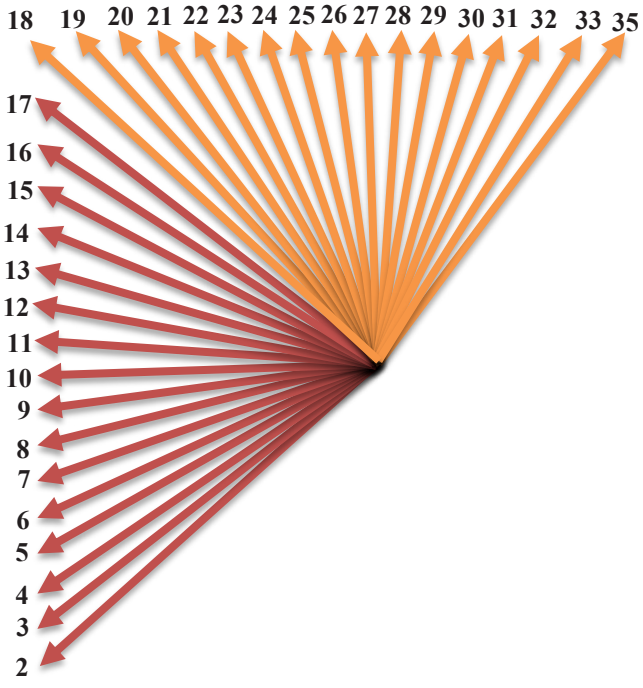

Figure 1: Intra modes in H265

Considering mode 7 as an example,

- Mode: 7
- Base Angle: 32 (a standard value in HEVC)
- Total Modes: 34
- Angle Calculation for Mode 7:

$$\text{actual\_angle} = \left(\frac{(7-2)\times 32}{34-2}\right) = 7$$

The reference position calculation is done as follows. For each pixel (y,x) in the block, the reference position (ref_{pos}) is calculated as:

$$ref_{pos} = \left[\frac{y * actual\_angle + x \times \text{base\_angle}}{\text{base\_angle}}\right]$$

For a 4x4 block the calculations are performed, assuming ref_left = [10, 12, 14, 16, 18]. The prediction values are shown in table 1.

The predicted block is shown in figure 2.

$$\begin{bmatrix} 10 & 12 & 14 & 16 \\ 10 & 12 & 14 & 16 \\ 12 & 12 & 14 & 14 \\ 14 & 14 & 14 & 16 \end{bmatrix}$$

Figure 2: predicted block

Table 1: Predicted value calculations

(y, x)	ref_pos calculation	ref_pos	Predicted Value (ref_left[ref_pos])
(0, 0)	(0 * 7 + 0 * 32) // 32 = 0	0	10
(0, 1)	(0 * 7 + 1 * 32) // 32 = 1	1	12
(0, 2)	(0 * 7 + 2 * 32) // 32 = 2	2	14
(0, 3)	(0 * 7 + 3 * 32) // 32 = 3	3	16
(1, 0)	(1 * 7 + 0 * 32) // 32 = 0	0	10
(1, 1)	(1 * 7 + 1 * 32) // 32 = 1	1	12
(1, 2)	(1 * 7 + 2 * 32) // 32 = 2	2	14
(1, 3)	(1 * 7 + 3 * 32) // 32 = 3	3	16
(2, 0)	(2 * 7 + 0 * 32) // 32 = 1	1	12
(2, 1)	(2 * 7 + 1 * 32) // 32 = 1	1	12
(2, 2)	(2 * 7 + 2 * 32) // 32 = 2	2	14
(2, 3)	(2 * 7 + 3 * 32) // 32 = 2	2	14
(3, 0)	(3 * 7 + 0 * 32) // 32 = 2	2	14
(3, 1)	(3 * 7 + 1 * 32) // 32 = 2	2	14
(3, 2)	(3 * 7 + 2 * 32) // 32 = 2	2	14
(3, 3)	(3 * 7 + 3 * 32) // 32 = 3	3	16

3.1.1 Edge Detection

Edge detection is an essential step of adaptive intra prediction, which is necessary to enhance the accuracy and efficiency of video compression. Recognizing whether there are edges and even their direction in a block of pixels allows the encoder to adaptively choose prediction modes that are more similar to the pixels in the image and, thus, generate more precise predictions and fewer residuals. That is, less data is encoded, which increases compression efficiency. Intra prediction is based on spatial redundancy within a frame and edges often correspond to abrupt changes in pixel values. Identifying these edges enables the prediction algorithm to enhance the final quality of compressed video. Thus, edge detection helps to maintain foreground and background edges in the image which otherwise may be lost during end-to-end compression leading theoretically to improved picture quality for high fidelity reconstruction of frames - particularly at lower bitrates.

In the edge detection module, 16 8-bit pixels processed to detect horizontal and vertical edges within a 4×4 pixel block. During each iteration of the pixel value, an exception of boundary pixels is given. Additionally, for each non-boundary pixel, the module computes a gradient in the x and y directions using the neighbouring pixel values. A simple convolution with a filter like Sobel is used. The horizontal gradient is a sum of pixel value differences along the row, and Vertical gradient sums over column. If the absolute value of gx or gy is greater than a threshold (128), then its corresponding edge detection flag, horizontal_edge and vertical_edge) respectively, are set to signal presence of an edge along that direction. This technique helps module to detect where are the edges present inside that pixel block and this can helpful for multiple image processing problems.

The algorithm steps are as follows:

Algorithm 1: Edge Detection

Step 1: Separate the 128-bit input into sixteen 8-bit pixel values.
Step 2: Initialize horizontal and vertical edge detection flags to zero.
Step 3: Iterate over the 16 pixel values.
Step 4: For each pixel, skip boundary pixels (i.e., first and last columns and rows).
Step 5: Calculate the horizontal gradient (gx) using neighboring pixel values.
Step 6: Calculate the vertical gradient (gy) using neighboring pixel values.
Step 7: Check if the absolute value of gx exceeds a threshold (128) and set the horizontal edge flag if true.
Step 8: Check if the absolute value of gy exceeds a threshold (128) and set the vertical edge flag if true.

3.1.2 Content Analysis

Content analysis is very important for adaptive intra prediction in video compression. This helps the encoder to better identify which prediction modes are appropriate for a block, by labeling them smooth or detailed based on their pixel variance. Simpler prediction modes could be used for smooth regions to lower computational complexity and data size. Dense pixel regions may use a more complicated mode to extract detailed textures, which will enable the prediction accuracy there and maintain visual quality.

The content analysis module is used to determine if the block of pixels is smooth or detailed. This module calculates spatial variance (i.e., pixel-by-pixel changes) within the block by summing the squared differences between each pixel and its immediate right neighbor as well as every bottom neighbor. This difference calculates the intensity differences between all possible pair of pixels within that block. If the total variance is lower than a threshold then the pixel block is classified as smooth. A large variance in the pixel block is classified as detailed block with high variance and textures. They improve the compression efficiency and even image quality by streamlining how blocks of specific types (e.g., flat regions, edges) are encoded.

Algorithm 2: Content Analysis

Step 1: Separate the 128-bit input into sixteen 8-bit pixel values.

Step 2: Initialize smooth and detailed flags to zero.

Step 3: Initialize the variance accumulator to zero.

Step 4: Iterate over the 16 pixel values.

Step 5: For each pixel, skip boundary pixels (those in the last column and last row).

Step 6: Calculate the variance by summing the squared differences between each pixel and its right and bottom neighbors.

Step 7: Check if the accumulated variance is below a threshold (1000) and set the smooth flag if true.

Step 8: Check if the accumulated variance is above a threshold (5000) and set the detailed flag if true.

The adaptive intra prediction module is optimized not only for high prediction accuracy but also to minimize power consumption. Power consumption is thus minimized by activating only the required prediction modes that are based on edge detection and content analysis. This filtering allows only the most informative predictions to be computed saving power in encoding due to selective processing.

3.2 Hierarchical Motion Estimation

Hierarchical motion estimation block creates a multi-resolution search to find an 8x8 matching area for each pixel group of interest (4pixels) within and other larger reference frame. First the 4x4 current frame block and reference frame are reshaped back into a searchable 2D array from flat. It has three resolution stages, full resolution followed by half resolution and finally quarter. In the first stage, it calculates SAD (Sum of Absolute Differences) between current block with all possible position on reference frame and record position that has minimum SAD. Then it downsamples the best match region and current block, then repeat SAD computation on those. Lastly, the process is repeated at quarter resolution to enhance precision. During all these steps, the module keeps track of best match coordinates with minimum SAD and perfect motion estimation to be used for video compression applications.

Algorithm 3: Hierarchical Motion Estimation

Initialize: Unpack the 128-bit current frame and 512-bit reference frame into 4x4 and 8x8 2D arrays, respectively.
Reset: On reset, set best_x, best_y to 0, and min_sad to the maximum possible value.
Full Resolution SAD Calculation: Iterate over possible positions in the 8x8 reference frame (limited to a 5x5 search window).
Compute SAD: For each position, calculate the Sum of Absolute Differences (SAD) between the 4x4 current block and the corresponding 4x4 block in the reference frame.
Update Best Match: If the current SAD is less than min_sad, update min_sad, best_x, and best_y with the current position and SAD.
Half Resolution Downsample: Downsample the current frame and the best matching region of the reference frame to 2x2 blocks by averaging groups of four pixels.
Half Resolution SAD Calculation: Iterate over possible positions in the downsampled 2x2 reference frame (within the previously found best match region).
Compute SAD: For each position, calculate the SAD between the 2x2 current block and the corresponding 2x2 block in the downsampled reference frame.
Update Best Match: If the current SAD is less than min_sad, update min_sad, best_x, and best_y with the refined position and SAD.
Quarter Resolution Downsample: Downsample the 2x2 blocks to a 1x1 block by averaging all four pixels.
Quarter Resolution SAD Calculation: Compute the SAD between the 1x1 current block and the 1x1 block in the downsampled reference frame.
Update Best Match: If the current SAD is less than min_sad, update min_sad, best_x, and best_y with the final refined position and SAD.
Output: Provide the final best_x and best_y coordinates as the position of the best match.

The hierarchical motion estimation block also contains low power capabilities. The system uses a multi-resolution search approach in servicing coarse-to-fine motion estimation to keep the computation reduced at every resolution. Low power implementation permits substantial power savings, since the most computationally demanding fine resolution calculations are only made to a small fraction of motion vectors inferred at coarser levels.

Low-power techniques are integrated into multiple levels of the proposed model to ensure efficient power management while preserving high level of performance and compression efficiency including edge detection, content analysis and hierarchical motion estimation. This makes the model ideal for edge and IoT devices, as well as for applications where energy usage comes under a premium.

3.3 Intelligent Clock Gating (ICG) Technique

Intelligent Clock Gating (ICG) is power saving technique used in the implementation of H.265 encoding in SoC. For ICG, the idea is to turn off a portion of the FPGA when it is no longer in use. This is performed by turning off clock signal, thereby reducing the dynamic power consumption which for high-speed SoCs and FPGAs. This method contributes significantly to he total operating power. The encoding for H.265 consists of multiple computational modules such as intra-frame prediction, motion estimation and entropy coding.

Some modules might not have to run at the same time, so for example during intra-frame prediction we may not need motion estimation module. Using ICG to clock gate the idle modules can effectively reduce switching power. This is especially advantageous for H.265 encoding, given the nature of its adaptive and hierarchical structures where the computational load depends not only on the video content being encoded but also on selected prediction modes themselves.

Ensuring that only the necessary part of a circuit (such as an adder, multiplier or DSP unit) is powered up at any given instant reduces static power consumption while maintaining high performance; this technique has been named ICG and drastically improve energy efficiency. ICG implementation in SoC-based H.265 encoders can strongly improve battery life of portable devices and reduce thermal footprint, thus forming an indispensable low-power - high-performance solution to video encoding.

4. Experimental Results

This section describes the simulation results of adaptive Intra Frame and Hierarchical Motion Estimation in H.265 video codec. The simulation results are discussed in two parts: in the first part intra prediction model's results are discussed and inter prediction model results are discussed in second part. The Adaptive Intra Frame and Hierarchical Motion Estimation in H.265 video codec is proposed to make the frame, inter/intra prediction & motion estimation process better so that high-quality compressed videos are achieved as compared to MPEG-2/H.263, H.264 encoding structure. Proposed adaptive Intra Frame prediction selects the optimal mode per region in a video frame, resulting in better balance between compression and picture quality. In the predicting motion estimation, the predicted changes are based on the order of predicted references, this complexity is addressed via multi-level approach by Hierarchical Motion Estimation. Individually, these features in the H.265 codec suggest a huge increase in compression rate and perceptual visual quality of the video encoded with this standard where quality of the videos are not compromised. The intra prediction model is developed based on adaptive framework. The modes are selected based on the pixel properties namely edges and content. The intra prediction model is discussed in section- 4.1

4.1 Intra Prediction model

The intra prediction model in implemented H.265 video codec uses adaptive techniques to improve the performance of pixel value predictability within a single frame. The model chooses the best suited prediction mode near to that content characteristics like smooth areas, horizontal edges or vertical ones. The intra-prediction driven through surrounding reference pixels to predict pixel values and then it helps in reduction spatial redundancy which means that proposed compression will be more beneficial. The model entails different modes like planar, DCs and angular predictions which are specialized for a particular type of texture or structure in the image. These characteristics make the codec flexible enough it can be used for compressing high-definition video without increasing bandwidth, therefore making way for better quality videos while using less space.)Fig. 3)

The simulation results shown in the Figure 3 which depict selection of 0 and 1 modes (SMOOTH areas) by implementing an intra prediction model for H.265 video codec.

Mode 0 (Planar Prediction): At the time interval from 0 to 100 ns, mode is set as 0 indicating planar prediction. The reference pixels used are "*000102030405060708090A0B0C0D0E0F*" representing a

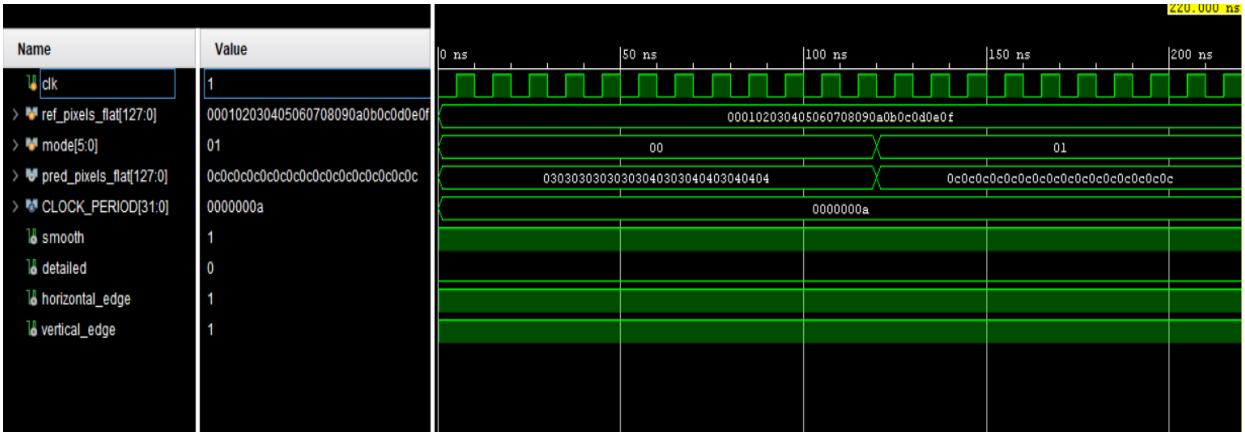

Fig. 3: Simulation waveforms for mode section 0 and 1 (smooth region)

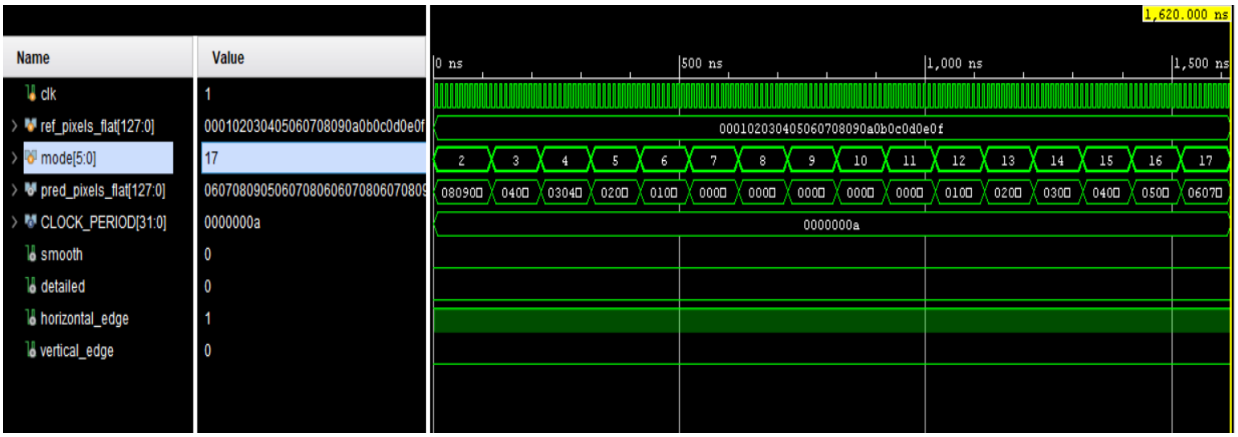

Fig. 4: Simulation waveforms for mode section from 2 to 17 (horizontal edge

sequence of 8-bit values for 4x4 block. During this time the obtained predicted pixels (*pred_pixels_flat*) are *"0 3030303030303040303040403040404"* and it denotes the smoothing of reference pixels for a shallower gradient across block, especially good at textures, making it picture-like calculation planar prediction mode.

Mode 1(DC Prediction): At the time duration 100 to 200 ns, a mode change operation is performed, the mode becoming (1) as described for DC prediction (mode). The reference pixels are the same as in direct encoding mode. The predicted pixels in these cases predict the mode transition as *"0C0C0C0C0C0C0C0C0C0C0C0C0C0C0C"*. DC prediction mode uses the average value of reference pixels which, when averaged and applied t0o all predicted pixels can appropriately deal with smooth areas across frame((Fig. 4).

Simulation waveforms of the intra prediction model for 2 to 17 modes selection in H.265 video codec is shown in the figure 4. These modes are meant to predict various angles and are specifically aimed at horizontal edges. Simulation waveforms showing the reacting of intra prediction model to different angular prediction modes. The prediction algorithm is adapted to the characteristics of the content for each mode, in this case horizontally edges. The pixel values were different from each other based one mode to another, which means that the model is able to predict in a better way how texture and structure are distributed in the window frame. These results confirm the flexibility and performance of this intra prediction model to improve video compression through precisely predicting pixel values by using spatial properties of content.

The modes from 18 to 34 are to predict angular in different prediction direction and mainly designed for dealing with vertical edges. This is followed by simulation waveforms depicted in figure 5, that show how the intra prediction model reacts to vertical edge-directed angular predictions modes. Each of these modes causes the predictor algorithm to act on this model, adjusting it according for vertical edges in one case. The changing of predicted pixels at each mode is represented in Table 2.

The table 2 provides predicted pixels, which will change according to the selected mode and inject different

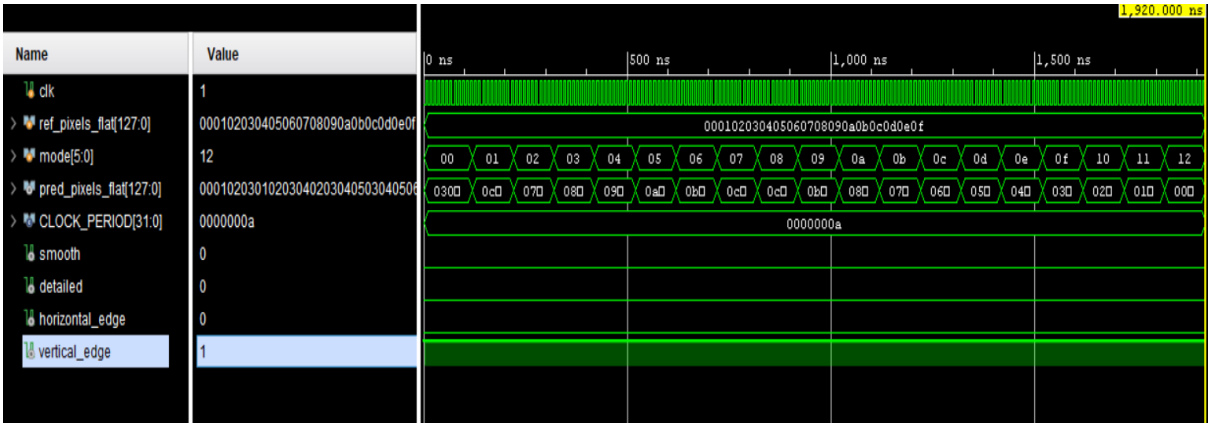

Fig. 5: Simulation waveforms for mode section from 18 to 34 (vertical edge)

Table 2: Prediction Pixel values at different modes

Reference Pixel variable name: *ref_pixels_flat*	Mode	Predicted Pixel variable name: *pred_pixels_flat*
000102030405060708090A0B0C0D0E0F	18	000102030102030401020304020304 05
	19	020304050203040502030405020304 06
	20	040506070405060704050607040506 08
	21	060708090607080906070809060708 0A
	22	08090A0B08090A0B08090A0B08090A 0C
	23	0A0B0C0D0A0B0C0D0A0B0C0D0A0B0C 0E
	24	0C0D0E0F0C0D0E0F0C0D0E0F0C0D0E 10
	25	0E0F0E0F0E0F0E0F0E0F0E0F0E0F0E 11
	26	0F0F0F0F0F0F0F0F0F0F0F0F0F0F 12
	27	0F0E0F0E0F0E0F0E0F0E0F0E0F0E 13
	28	0E0D0E0D0E0D0E0D0E0D0E0D0E0D 14
	29	0D0C0D0C0D0C0D0C0D0C0D0C0D0C 15
	30	0C0B0C0B0C0B0C0B0C0B0C0B0C0B 16
	31	0B0A0B0A0B0A0B0A0B0A0B0A0B0A 17
	32	0A090A090A090A090A090A090A09 18
	33	090809080908090809080908090819
	34	080708070807080708070807080720

angular relationships inside reference pixel block into their contexts. By focusing on a particular angular pattern, each mode helps the model predict pixel values that mimic what it found in textures and edges of the video frame. For this case, vertical edges are highlighted in modes where there is the most significant increase on top of progressive pixel values, and this refers to how well vertical structures will be recognized by each model. This adaptation serves to maintain the visual reliability of video by keeping its vertical texture intact. The different modes and their corresponding changes allowed the model to accurately predict what pixel values for smooth regions in between, as well as areas with vertical patterns. This versatile prediction scheme leads to a more accurate encoding of the video and improves both coding efficiency, as well as visual

experience which makes H.265 codec reliable for all kind of videos by using this adaptive ability.

4.2 Inter Prediction model

Inter Prediction model in H.265 video codec is a sophisticated approach to improve the compression efficiency of videos by exploiting on temporal redundancies between consecutive frames. This works by predicting the content of a frame based on one or many reference frames, i.e. it reduces data needed to represent motion and changes over time in next video frames. This Inter Prediction model features sophisticated motion estimation and compensation techniques to find the changes between frames, which results in substantial data reduction. A hierarchical motion estimation then works along a coarse-to-fine solution which aims to increase the accuracy of movement vectors, crucial for the proper prediction and objects reduction. The model excels especially at sequences with a lot of motion, and it yields an optimal compression rate that best achieves good visual quality. The simulation results of the proposed inter prediction model is depicted in Figure 6.

This figure 6 shows the simulation waveforms that describe how the proposed hierarchical motion estimation module works in a H.265 coded video stream. This simulation mainly deals with a more focused approach to find the best possible match for the current frame block within reference frame hierarchical motion estimation. A consistent period is shown on the clk signal, to indicate synchronized behaviour over the course of a simulation. The reset signal is high at the beginning for resetting module and then set to low. This transition enables to module for processing input frames as well doing motion estimation.

There are three steps in the hierarchical motion estimation process: full resolution, half resolution and quarter resolution motion. During the full resolution step, the lowest SAD for each candidate block in a reference frame is calculated by this module and in case of receiving lower SAD, it will also update best coordinates. These coordinates are then further fine-tuned with half and quarter resolution steps: taking averages of pixel values, searching for small regions. The outputs *best_x* and *best_y* depict the *x* and *y* coordinates of the best match among the reference frames. As can be seen from the figure, the values of both *best_x* and *best_y* are at a stable point of 4. which means that the location for the best matching block in the reference frame for the current frame is precisely at 4x4. These coordinates were achieved from the developed module under a hierarchical search, with the priority of minimizing Sum of Absolute Difference (SAD) between the current frame block and several candidates blocks on the reference frame.

4.3 comparative analysis

In addition to the differences in algorithm, it is important to make a comparative analysis of the proposed Inter and Intra prediction models with respect previous works for evaluating their performance gain as well efficiency against existing work on H.265 video codec. This analysis will identify the changes in computational complexity (resource utilization in SoC). The resource utilization reports are taken into consideration to evaluate the proposed model. One of the most critical aspects that drives efficiency, performance and scalability of integrated circuits is resource utilization in SoC design. Involving Optimization of the use of all available resources like processing cores, memory blocks, I/O interfaces and

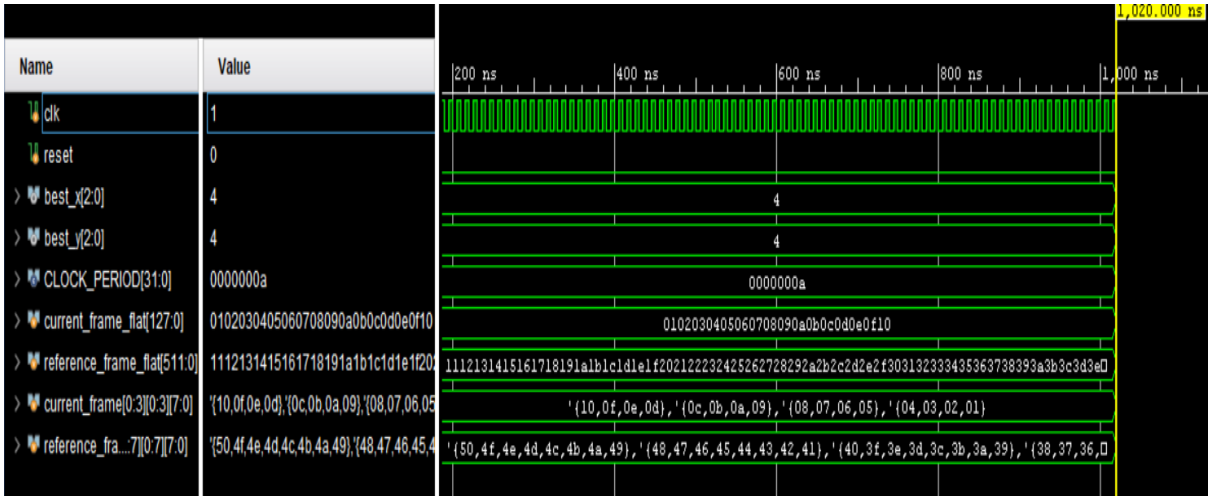

Fig. 6: Simulation results of the proposed inter prediction model with hierarchical motion estimation

specialized accelerators on SoC. This optimization helps the SoC to process complex computation needs with low-power budget and high-performance. The resource utilization of proposed Intra prediction with existing model is reported in Table 3.

The table 3 gives a quantitative comparison of the resource consumption metrics between proposed Intra model and existing model. Metrics include Slice LUTs, Logic LUTs, Slice Registers, Block RAMs and Bounded IOBs. The results are presented according to Smooth, Horizontal and Vertical with different prediction types. The proposed model utilizes much less Slice LUTs compared to the existing model. Comparatively, the proposed Intra model is resource-efficient with respect to the existing. It uses less Slice LUTs and Logic LUTs. This is a clear indication that the proposed model not only increases capabilities, but also ensures optimal usage of resources. The resource utilization of proposed Inter prediction with existing model is reported in Table 4.

The table 4 shows the comparison of resource utilization metrics between existing inter prediction model and proposed model. These metrics include Slice LUTs, Slide Registers and Bounded IOBs. The proposed model on the other hand shows a remarkable reduction in Slice LUT utilization with only 233 replacing the existing model's use of 7459. This significant reduction emphasized the effectiveness of the proposed model regarding logic implementation and could possibly help in reducing power consumption as well improving overall performance by decreasing the complexity of logic. The proposed model uses 40 Slice Registers a little more than the existing model (19). This implies that the proposed new model is added with more number of flip-flops or registers to improve its performance as it spends some additional resource count in this particular metric. With only 8 Bounded IOBs used in the proposed model out of 648 with existing. As a result, the reduces I/O resources which can in turn be utilized to interface with various external entities more efficiently and might save on total power consumption which is an important aspect of real-world applications deployment aimed at battery life environment.

Table 5 presents the power report of the proposed intra prediction model. The dynamic power of the proposed model using ICG is 0.893 W, whereas the existing models power utilization is 1.967 W. The static power utilised by the proposed model is also less which is equal to 0.121W, whereas the static power of the conventional model is 0.149 W The total on chip power of the proposed model is 1.14W whereas the conventional method is 2.116W.

Figure 7 shows the on chip power report produced by the proposed model.

Table 6 shows the power report of the proposed inter prediction model that is hierarchical motion estimation. The proposed model with ICG obtain the dynamic power, static power and total on chip power as 1.025 Watt, 0.124 Watt and 1.149 Watt respectively. Whereas the conventional methodology using ICG produced a dynamic power, static power and total launching power as 1.934, 0.148 and 2.082 respectively.

Table 3: Comparison results of Resource utilization (Intra Prediction)

Metric	Proposed Model			Existing model 17
	Smooth	Horizontal	Vertical	
Slice LUTs	228	1069	1355	1969
Logic LUTs	228	1069	1355	1969
Slice Registers	0	0	0	0
Block RAMs	0	0	0	0
Bounded IOB	262	262	262	262

Table 4: Comparison results of Resource utilization (Iner Prediction)

Metric	Existing Model [18]	Proposed Model
Slice LUTs	7459	233
Slice Registers	19	40
Bounded IOB	648	8

Table 5: Power report of Proposed Intra prediction model

	Power Utilization of Intra Prediction model without using ICG (W) [17]	Power Utilization of Proposed Intra Prediction model using ICG (W)
Dynamic Power	1.967	0.893
Static Power	0.149	0.121
Total On Chip power	2.116	1.014

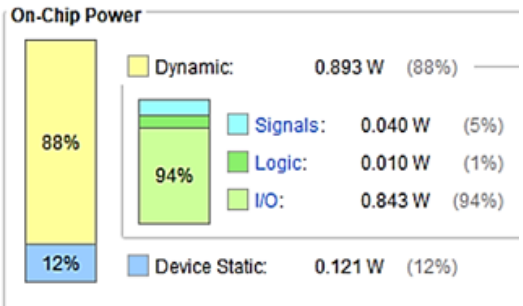

Fig. 7: On chip power report Intra prediction

Table 6: Power report of Proposed Inter prediction model

Metric	Power Utilization of Inter Prediction model without using ICG (W) [18]	Power Utilization of Proposed Inter Prediction model using ICG (W)
Dynamic Power	1.934	1.025
Static Power	0.148	0.124
Total On Chip power	**2.082**	**1.149**

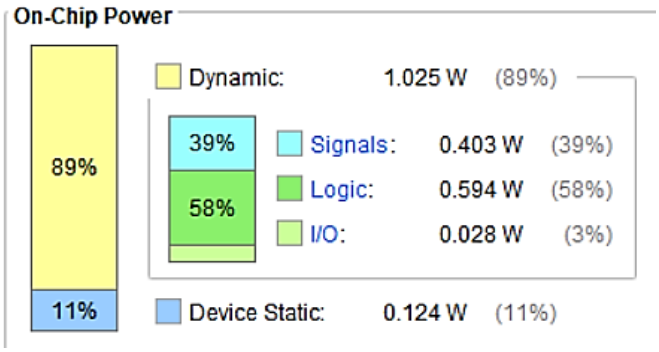

Fig. 8: On chip power – inter prediction motion estimatio

Figure 8 shows the on chip power of the proposed motion estimation framework. These results clearly present the low power nature of the proposed framework in comparison to the conventional techniques.

5. Conclusion

The key contributions and findings were that the adaptive intra-frame and hierarchical motion estimation achieve the most substantial rise in the H.265 codec efficiency. Adaptive intra-frame prediction utilizes the best allocation of modes and directions of the corresponding pixels. Hierarchical motion estimation reduces the errors in the motion vector prediction. Due to this, the video quality is high as the computational or other resources put into compression are low. Through integration of edge detection, content analysis and Sum of Absolute Differences (SAD) algorithms the model optimizes compression rates to meet the set standards yet maintaining visual quality. The proposed methodology overcomes the heavy computational load that H.265 encoding imposes, making it difficult to achieve real-time performance in resource-limited devices. This research has important consequences, demonstrating that the model can achieve a remarkable improvement in terms of computational and memory resource cost at such compression efficiency levels.

References

1. Jia, Chuanmin, Xinyu Hang, Shanshe Wang, Yaqiang Wu, Siwei Ma, and Wen Gao. "Fpx-nic: An fpga-accelerated 4k ultra-high-definition neural video coding system." *IEEE Transact ions o n Circuits and Systems for Video Technology* 32, no. 9 (2022): 6385-6399.
2. Kurshid, Bimer, et al. "The Potential of Ultra-Wideband Printed Rectangular-Based Monopole Antennas." *National Journal of Antennas and Propagation* 5.2 (2023): 14-20
3. Chatterjee, Sumit Kumar, and Sravan Kumar Vittapu. "FPGA implementation of EFSME for high efficient video coding standard." *Multimedia Tools and Applications* 81, no. 23 (2022): 34087-34103.
4. Chettri, Lalit, and Rabindranath Bera. "A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems."IEEE Internet of Things Journal7.1 (2019): 16-32
5. Chatterjee, Sumit Kumar, and Sravan Kumar Vittapu. "FPGA implementation of EFSME for high efficient video coding standard." *Multimedia Tools and Applications* 81, no. 23 (2022): 34087-34103.
6. Saadawi, Enas Magdi, Abdelaziz Said Abohamama, and Mohammed Fathi Alrahmawy. "IoT-based Optimal Energy Management in Smart Homes using Harmony Search Optimization Technique." (2022).
7. Zhang, Ying, Gen Li, and Lei Wang. "A High-Performance with Low-Resource Utility FPGA Implementation of Variable Size HEVC 2D-DCT Transform." In *Advanced Computer Architecture: 13th Conference, ACA 2020, Kunming, China, August 13-15, 2020, Proceedings 13*, pp. 325-333. Springer Singapore, 2020.
8. Zaki, Farid, Amr E. Mohamed, and Samir G. Sayed. "CtuNet: A deep learning-based framework for fast CTU partitioning of H265/HEVC intra-coding." *Ain Shams Engineering Journal* 12, no. 2 (2021): 1859-1866.
9. Javaid, Mohd, et al. "5G technology for healthcare: Features, serviceable pillars, and applications."Intelligent Pharmacy(2023)
10 Kopperundevi, P., Matcha Surya Prakash, and Shaik Rafi Ahamed. "A high throughput hardware architecture for deblocking filter in HEVC." *Signal Processing: Image Communication* 100 (2022): 116517.
11. Uvarajan, K. P., and K. Usha. "Implement A System For Crop Selection And Yield Prediction Using Random Forest Algorithm." *International Journal of communication and computer Technologies* 12.1 (2024): 21-26.
12 Sheng, Qingxin, Chong Fu, Ming Tie, Xingwei Wang, Junxin Chen, and Chiu-Wing Sham. "A Chaos-based Tunable Selective Encryption Algorithm for H. 265/HEVC with Semantic Understanding." *IEEE Transactions on Circuits and Systems for Video Technology* (2024).
13. Zaki, Farid, Amr E. Mohamed, and Samir G. Sayed. "CtuNet: A deep learning-based framework for fast CTU partitioning of H265/HEVC intra-coding." *Ain Shams Engineering Journal* 12, no. 2 (2021): 1859-1866.

14. Wei, Lee, and Wai Cheng Lau. "Modelling the Power of RFID Antennas By Enabling Connectivity Beyond Limits." *National Journal of Antennas and Propagation* 5.2 (2023): 43-48

15. Pastuszak, Grzegorz. "Multisymbol architecture of the entropy coder for H. 265/HEVC video encoders." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 28, no. 12 (2020): 2573-2583.

16. Premakumari, R. N., et al. "Modeling the dynamics of a marine system using the fractional order approach to assess its susceptibility to global warming." Results in Nonlinear Analysis 7.1 (2024): 89-109.

17. Gogoi, Sushanta, and Rangababu Peesapati. "A hybrid hardware oriented motion estimation algorithm for HEVC/H. 265." *Journal of Real-Time Image Processing* 18, no. 3 (2021): 953-966.

18. Gogoi, Sushanta, and Rangababu Peesapati. "Design and implementation of an efficient multi-pattern motion estimation search algorithm for HEVC/H. 265." *IEEE Transactions on Consumer Electronics* 67, no. 4 (2021): 319-328.

19. George, Reny, et al. "Some existential fixed point results in metric spaces equipped with a Graph and it's application." *Results in Nonlinear Analysis* 7.1 (2024): 122-141.

20. Ni, Chi-Ting, Ying-Chia Huang, and Pei-Yin Chen. "A hardware-friendlyand high-efficiency H. 265/HEVC encoder for visual sensor networks." *Sensors* 23, no. 5 (2023): 2625.

21. Alarifi, Abdulaziz, Syam Sankar, Torki Altameem, K. C. Jithin, Mohammed Amoon, and Walid El-Shafai. "A novel hybrid cryptosystem for secure streaming of high efficiency H. 265 compressed videos in IoT multimedia applications." *IEEE Access* 8 (2020): 128548-128573.

22. Park, Sehie. "All metric fixed point theorems hold for quasi-metric spaces." *Results in Nonlinear Analysis* 6.4 (2023): 116-127.

23. Pham-Quoc, Cuong. "FPGA-Based Hardware/Software Codesign for Video Encoder on IoT Edge Platforms." In *International Conference on Computational Science and Its Applications*, pp. 82-96. Cham: Springer Nature Switzerland, 2023.

24. Steinert, Fritjof, and Benno Stabernack. "Architecture of a Low Latency H. 264/AVC Video Codec for Robust ML based Image Classification: How Region of Interests can Minimize the Impact of Coding Artifacts." *Journal of Signal Processing Systems* 94, no. 7 (2022): 693-708.

25. Kankarej, Manisha, and Jai Pratap Singh. "Fractional Curl With Standard Fractional Vector Cross Product For A Vector Pair: Fractional Curl With Standard Fractional Vector Cross Product For A Vector Pair." *Results in Nonlinear Analysis* 6.3 (2023): 19-29.

26. Lu, Yufan, Cong Gao, Rappy Saha, Sangeet Saha, Klaus D. McDonald-Maier, and Xiaojun Zhai. "FPGA-based dynamic deep learning acceleration for real-time video analytics." In *International Conference on Architecture of Computing Systems*, pp. 68-82. Cham: Springer International Publishing, 2022.

27. Lam, Duc Khai, Pham The Anh Nguyen, and Tuan Anh Tran. "Hardware Architecture for Realtime HEVC Intra Prediction." *Electronics* 12, no. 7 (2023): 1705. *DOI:* https://doi.org/10.3390/electronics12071705 .

28. Loukil, Hassen, and Abdulilah Mohammad Mayet. "Hardware implementation and validation of the fast variable block size motion estimation architecture for HEVC Standard." *Multimedia Tools and Applications* 82, no. 30 (2023): 46331-46349.