DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

IPMV Project – Continuous Assessment

Third Year/Semester-VI/2023-24

PROJECT NAME		**Creating Simulink based simulation for Run-length Coding**
Group Number		**17**
Member 1		Aditi Boob (D14B/08)
Member 2		Anjali Gupta (D14B/21)
Member 3		Neerav Sonaje(D14B/60)
Member 4		Aditi Ullas(D14B/65)
Name of the Supervisor		**Prof. Mrugendra Vasmatkar**
Sem/Year/CAY		VI/TE/2023-24
Define the problem & relevance to today's market/society/industry need (Max 100 words)	The design of a Simulink model for implementing run-length coding (RLC) on a binary input signal. The model should comprise blocks that detect consecutive identical symbols, count their occurrences, and produce the encoded output sequence. The primary objective is to simulate the model to observe the RLC encoding process and evaluate its efficacy in compressing the binary input signal, thereby assessing the performance of the RLC algorithm within the Simulink environment.	
Propose the solution to Problem Identified (Max 100 words)	A proposed solution for implementing run length coding (RLC) in Simulink involves designing a model that takes input data sequences and generates corresponding encoded sequences using the RLC algorithm. Here's a step-by-step approach to implementing this solution:effectively implement run length coding in Simulink and assess its performance for compressing data sequences. This approach leverages Simulink's graphical modeling capabilities and MATLAB's computational tools to create a comprehensive solution for RLC encoding and evaluation.	
OBJECTIVE(s)	**List the objectives of your project** 1) **Understanding RLC:** Implement run-length coding in MATLAB Simulink to compress a sequence of random numbers generated within the Simulink environment.	

	The goal is to design a system that efficiently encodes the input sequence using the run-length coding technique and visually displays both the original and encoded sequences for evaluation and verification purposes. 2) **Design and Simulate RLC on Simulink:** This objective involves creating a RLC model within Simulink, a simulation tool, to assess its performance and functionality 3) **Data Compression:** RLC is primarily used for data compression purposes. By encoding consecutive repeated values as a single value and count pair, RLC reduces the amount of data required to represent the original information. This compression can be particularly beneficial when dealing with large datasets or when transmitting data over bandwidth-limited channels 4) **Simplified Signal Processing:** RLC can simplify signal processing tasks by reducing the complexity of the data being processed. This can lead to faster processing times and reduced computational resources required for signal analysis or manipulation
SPECIFIC:	The specific objective of implementing Run-Length Encoding (RLE) within Simulink is to develop a model that can efficiently compress data by identifying consecutive identical symbols in a binary input signal and encoding them as a single symbol followed by the count of occurrences. The Simulink model should accurately capture the RLE algorithm's functionality, including detecting consecutive identical symbols, counting them, and generating the encoded output sequence. The simulation aims to demonstrate the effectiveness of RLE in compressing binary data, thereby showcasing its utility for data compression applications within the Simulink environment.efficiency.Our Team includes four members Aditi Boob, Anjali Gupta, Neerav Sonje and Aditi Ullas.
MEASURABLE:	Our objective is to create a Simulink-based system for run-length coding (RLC) image compression. This entails designing and implementing an efficient compression algorithm within Simulink that utilizes RLE techniques to

	reduce the storage requirements of digital images while maintaining visual fidelity. By effectively encoding image data to eliminate redundancy, our system aims to achieve significant compression ratios without compromising image quality.
ACHIEVABLE:	The end result of this project will be a Simulink model specifically designed for run-length coding (RLC) image compression. This model will efficiently reduce the storage space required for digital images while maintaining their visual quality. Users will input images into the Simulink system, which will then perform RLC compression. Expertise in Simulink for simulation, problem-solving, and optimization skills are crucial for successful project execution.
RELEVANT:	RLC, or Run-Length Coding, is highly relevant in data compression applications where minimizing storage or transmission requirements is paramount. It finds extensive use in image and video compression algorithms, enabling significant reductions in file sizes while preserving visual quality. RLC also plays a crucial role in optimizing storage utilization in resource-constrained environments like embedded systems and cloud storage. Furthermore, it facilitates efficient data transmission by reducing bandwidth requirements, making it ideal for streaming multimedia content and communication systems. In signal processing tasks, RLC aids in analyzing and processing data more effectively, particularly in scenarios involving repetitive patterns or sequences. Its versatility and effectiveness make RLC a fundamental technique across various domains, enhancing efficiency and performance in data handling and communication
S.M.A.R.T. Goal	This goal centers on the creation of a MATLAB program aimed at enhancing the efficiency of image processing, particularly in terms of image size reduction. The objective is to develop a program capable of reducing the size of images by at least half while maintaining clarity and sharpness.

Introduction:

Run-Length Coding (RLC) is a key data compression technique, efficiently reducing redundancy by replacing repeated symbols with a single instance followed by a count. Widely applied in image processing, multimedia compression, and data transmission, RLC is effective for datasets with frequent consecutive identical symbols or long sequences of zeros. To enhance compression efficiency and facilitate faster data transmission, a Simulink-based simulation is proposed for RLE encoding and decoding. This versatile system aims to address traditional compression challenges, offering scalability and adaptability for various data types and applications. Leveraging Simulink's capabilities, the simulation will accurately emulate RLE algorithms, aiding research and application development in data compression.The Simulink-based simulation offers a robust platform for experimenting with different compression strategies and fine-tuning RLC algorithms to optimize performance. Its scalability allows for seamless integration with existing systems and potential extensions to handle evolving data compression requirements. By providing a realistic environment for testing and validation, this simulation contributes to advancements in data compression research and supports practical implementations in diverse domains.

Flowchart :

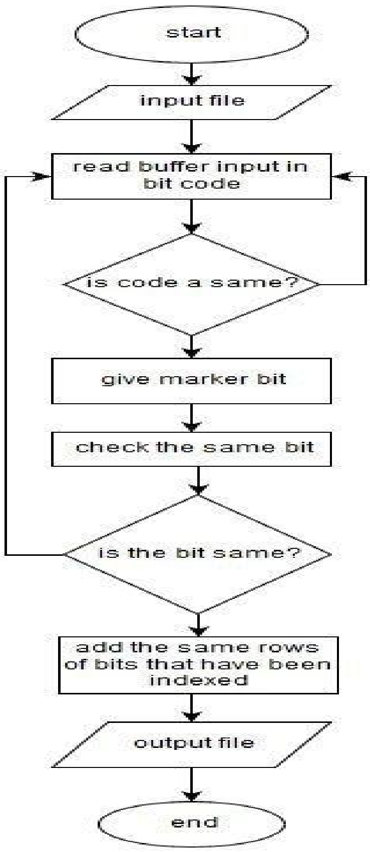

Fig 1: Flowchart of the proposed solution

1. Scan the data for sequences of identical characters exceeding a threshold (e.g., 3 characters). Focus on compressing longer repetitions.

2. Include a unique bit sequence (marker) to signal the beginning of compressed data within the encoded stream (improves clarity during decompression).

3. RLC replaces long stretches of identical characters with a count and the repeated character itself. This reduces the overall data size.

4. Decompression reads the count, replicates the character that many times, and reassembles the original data stream.

Block diagram:

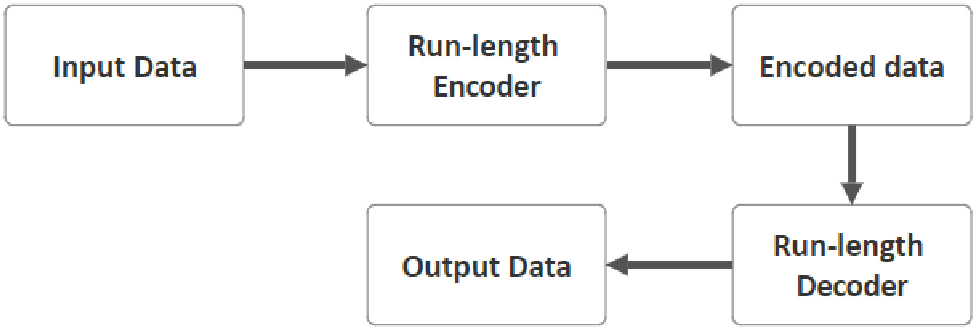

Fig 2: Block diagram of the proposed solution

1. **Input Data** : This block represents the original data sequence that needs to be compressed using the RLC algorithm. It could be any type of data, such as a binary sequence, image pixels, or text characters.

2. **Run-Length Encoder** : This block is responsible for compressing the input data by identifying consecutive identical symbols and replacing them with a single symbol followed by a count of their occurrences. The encoder generates the encoded data based on this compression scheme.

3. **Encoded Data** : This block represents the compressed data output by the Run-Length Encoder. It contains the compressed representation of the input data, typically consisting of pairs of symbols and their respective counts.

4. **Run-Length Decoder** : This block performs the inverse operation of the encoder. It takes the encoded data as input and reconstructs the original data sequence by expanding the compressed representations back into their original form.

5. Output Data : This block represents the reconstructed or decompressed data sequence generated by the Run-Length Decoder. It ideally matches the original input data, allowing for lossless data compression and decompression using the RLC algorithm.

Mentor Name & Signature with date