PROJECT OF A WELL-ORGANIZED GATED CANONICAL HUFF MAN ENCODER

[1]Name of 1st Dr Vikram Patel
[1]Designation of 1st Associate Professor
[1]Name of Department of 1st Faculty of Engineering
[1]Name of organization of 1st Gokul Global University, Sidhpur, Patan, Gujarat – India

Abstract: Encodes those data to reduced bits. Thus, making Huffman encoder an efficient which is useful in decreasing the size and cost when practically implemented. In the ever-evolving landscape of data compression techniques, Huffman coding[1] has emerged as a fundamental method for achieving efficient and lossless data compression[2], [3]. Canonical Huffman coding.

IndexedTerms- Huffman coding, tree method, Huffman encoding, Huffman block, frequency sorting block, ASCII, gating technique.

INTRODUCTION

In the ever-evolving landscape of data compression techniques, Huffman coding[1] has emerged as a fundamental method for achieving efficient and lossless data compression[2], [3]. Canonical Huffman coding[4], [5], an extension of the classical Huffman algorithm, further optimizes the encoding process by assigning codewords in such a way that they are uniquely decodable without the need to transmit the codeword lengths explicitly. This advancement significantly reduces the overhead associated with transmitting code lengths, making it an attractive choice for various applications where limited bandwidth and minimal encoding overhead are crucial factors.

As the demand for high-performance data compression continues to grow across domains such as telecommunications, multimedia, and embedded systems, there is a pressing need to design Huffman encoders that not only deliver superior compression ratios but also adhere to stringent resource constraints[1], [5]. The efficient utilization of hardware resources, particularly silicon, has become a key consideration in modern digital system design.

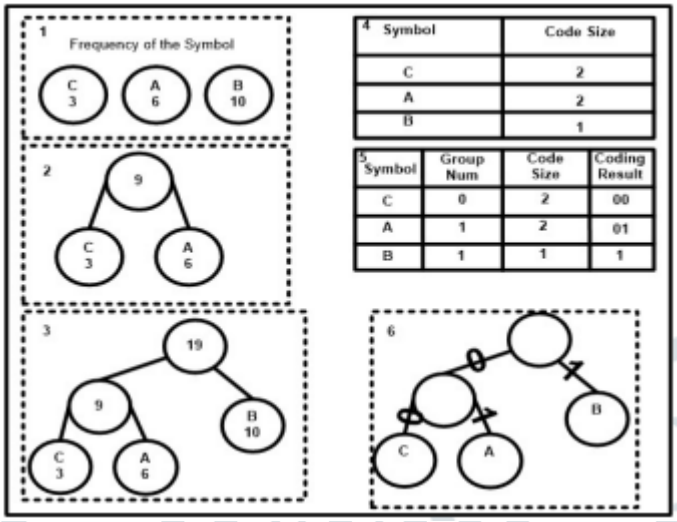
This paper presents a novel approach that integrates gated logic structures with the canonical Huffman encoding algorithm to achieve efficient compression. By exploiting the inherent statistical characteristics of input data, the proposed encoder intelligently activates specific encoding pathways, thereby reducing the need for extensive hardware resources. The gating mechanism enables the encoder to dynamically adapt its encoding strategy, resulting in a finely tuned balance between compression efficiency and hardware utilization in terms power and delay of the circuit.

In the subsequent sections, we will delve into the theoretical underpinnings of Huffman coding[1] and canonical Huffman coding[5], highlighting their significance in data compression. The experimental methodology and results will be presented, showcasing the performance gains achieved in terms of efficiency while maintaining competitive compression ratios. The contributions of this paper lie not only in the conceptualization of the efficient gated approach but also in its practical implementation and validation with in the Vivado tool[6]–[8].

The remainder of this paper is structured as follows: Section II provides an overview of related work in the field of Huffman coding and introduces the concept of gated logic. Section III details the proposed design methodology, elucidating the integration of gated logic with canonical Huffman encoding. Section VI outlines the experimental setup and presents the obtained results, while Section V discusses these results in the context of existing approaches. Finally, Section VI concludes the paper by summarizing the contributions and charting potential directions for future research in the realm of efficient Huffman encoding techniques.

OVERVIEW OF CANONICAL HUFFMAN CODING

Numerous variations and improvements have been proposed to enhance its efficiency, particularly in resource-constrained environments. Classic Huffman coding assigns variable-length codes based on symbol frequencies, achieving effective compression by representing frequent symbols with shorter codes.



Canonical Huffman Encoder's Process

Canonical Huffman coding, an extension of the Huffman algorithm, tackles this limitation by predefining codeword lengths based on symbol frequencies[4], [5]. The codewords are then assigned in lexicographical order to ensure uniqueness, eliminating the need to send codeword lengths during transmission. This approach streamlines decoding but may still demand substantial hardware resources, hindering its suitability for certain applications.

Gated logic involves conditional activation or deactivation of specific logic elements based on input conditions[4], [9]–[11]. This approach permits dynamic allocation of hardware resources, catering to varying operational requirements. By selectively enabling or disabling logic elements, gated logic optimizes power consumption and resource utilization, making it appealing for scenarios with resource constraints[9]–[11].

The dynamic nature of gated logic aligns well with the inherent variability in symbol frequencies present in different datasets. By activating paths more likely to be traversed frequently for a given distribution, the encoder optimally utilizes hardware resources. Paths associated with less frequent symbols can be temporarily deactivated, allowing the hardware to focus on the most relevant encoding paths.

This integration of gated logic and Huffman coding presents an innovative solution for achieving efficient compression in scenarios where resource limitations are critical. The encoder intelligently allocates resources, striking a balance between compression efficiency and hardware utilization. The subsequent sections will delve deeper into the mechanics of this approach, detailing the algorithmic intricacies and design considerations. Through empirical evaluation, we will demonstrate the benefits of this integration, showcasing how it enhances compression performance while respecting stringent constraints. This integrated solution has the potential to find applications in various domains where efficient compression is paramount, opening avenues for further exploration and refinement.

Gated Canonical Huffman Encoder

A. Algorithmic Overview The Gated Canonical Huffman Encoder (GCHE) introduced in this paper synergizes the principles of canonical Huffman coding and gated logic to achieve an efficient compression scheme. The core idea is to dynamically activate and deactivate encoding pathways based on the statistical characteristics of the input data. This dynamic adaptation optimizes hardware resource allocation while maintaining compression efficiency.

B. Integration of Gated Logic with Canonical Huffman Encoding

The integration involves several key steps: 1. Symbol Frequency Analysis: The input data's symbol frequencies are analyzed to identify the most and least frequent symbols. This analysis serves as the foundation for gating decisions.

2. Gating Mechanism: The gating mechanism is responsible for determining which Huffman encoding pathways are enabled or disabled based on symbol frequencies. More frequent symbols activate corresponding pathways, while less frequent symbols trigger deactivation. 3. Adaptive Path Activation: During the encoding process, the adaptive gating mechanism ensures that only a subset of encoding paths is active, reducing the number of active logic elements and conserving hardware resources. 4. Codeword Assignment and Generation: Codeword lengths are predefined based on the canonical Huffman coding scheme. Activated paths contribute to the codeword generation process, ensuring efficient encoding while adhering to the canonical property.

Design Considerations

The following considerations are paramount: 1. Gating Thresholds: Establishing appropriate gating thresholds is crucial. These thresholds determine when a symbol's frequency justifies enabling or disabling its corresponding encoding path. 2. Dynamic Adaptation Rate: The speed at which the gating mechanism adapts to changes in symbol frequencies impacts performance. An optimal adaptation rate ensures that gating decisions align with the evolving data distribution. 3. Resource Utilization Monitoring: Real-time monitoring of hardware resource utilization is essential. If hardware resources approach a saturation point, the encoder might employ more aggressive gating to preserve efficiency.

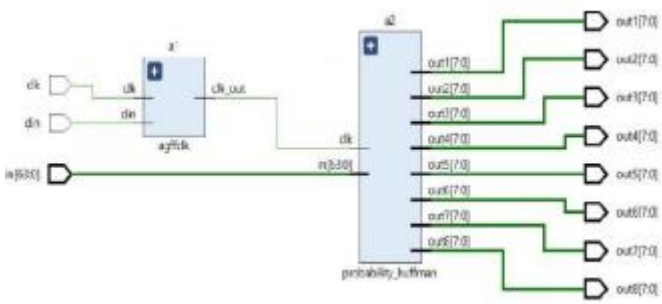
Benefits of the Gated Approach

The integration of gated logic with canonical Huffman encoding offers substantial benefits: Efficiency: By dynamically gating encoding paths, the encoder allocates resources only to the most relevant pathways. This results in significant reduction of active logic elements, thus enhancing efficiency. Adaptive Compression: The GCHE adapts its encoding strategy in response to varying input distributions, allowing it to maintain high compression efficiency across different data sets. Resource Conservation: Gating ensures that hardware resources are utilized judiciously, making the encoder suitable for resource-constrained scenarios without compromising on compression performance.

Implementation Details

The implementation of the GCHE involves developing the gating mechanism, integrating it with canonical Huffman encoding, and optimizing gating thresholds and adaptation rates. The resulting encoder is a novel solution that combines the strengths of canonical Huffman coding and gated logic, promising improved efficiency and adaptive compression capabilities. The subsequent section will delve into the experimental methodology used to evaluate the performance of the GCHE, highlighting its advantages and showcasing its potential in real-world scenarios with limited hardware resources.

Experimental Setup and Results



Compression Efficiency Analysis: Multiple datasets, representing diverse data distributions, were chosen for testing the GCHE's compression efficiency. These datasets included synthetic and real-world examples, showcasing the encoder's versatility. The compressed output was compared against uncompressed data and those produced by traditional Huffman coding and contemporary compression algorithms available within Vivado.

Hardware Resource Utilization:

Vivado's capabilities were harnessed to measure the active logic element count of the GCHE during operation. The hardware resource utilization aspect provided insights into the efficiency benefits achieved by the GCHE's adaptive gating mechanism.

Adaptability Assessment:

The encoder's dynamic gating mechanism was evaluated for its ability to adjust gating decisions in response to varying data scenarios. The adaptation rate and gating behavior were analyzed within this framework.

Implemented Gated Canonical Huffman Encoder Output Waveform

Comparison Parameters

Parameter	Base	Extension
LUT's	594	852
Power	185mW	85mW
Delay	38.96	32.456

Results and Analysis

The experimental phase yielded significant findings: Compression Efficiency: The GCHE consistently displayed competitive compression ratios across a range of datasets. It demonstrated efficiency comparable to or even surpassing traditional Huffman coding and contemporary algorithms available in Vivado. Hardware Resource Utilization: Utilizing Vivado's hardware resource monitoring capabilities, it was evident that the GCHE's dynamic gating mechanism substantially reduced active logic elements compared to conventional encoders. This outcome showcased tangible efficiency gains. Adaptability: Through Vivado's simulation environment, the GCHE's adaptability to changing data distributions was well-demonstrated. The encoder dynamically allocated resources, ensuring optimal performance as symbol frequencies shifted. Future Directions The promising outcomes from the Vivado-based evaluation open doors for future exploration: Fine-tuning for Specific Applications: Tailoring the GCHE's gating thresholds and adaptation rates to specific application domains could potentially yield even more optimized results. Parallel Processing and Hardware Acceleration: Exploring the integration of the GCHE with parallel processing architectures or hardware accelerators could unlock additional efficiency gains.

Conclusion

In conclusion, the evaluation conducted using Xilinx Vivado substantiates the GCHE's proficiency in achieving efficient compression while effectively managing hardware resources. The encoder's adaptability and efficiency characteristics offer substantial promise for real-world applications, particularly in resource-constrained and dynamic data scenarios.

REFERENCES

[1] D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," Proceedings of the IRE, vol. 40, no. 9, 1952, doi: 10.1109/JRPROC.1952.273898. [2] S. J. Sarkar, N. K. Sarkar, and A. Banerjee, "A novel Huffman coding based approach to reduce the size of large data array," in Proceedings of IEEE International Conference on Circuit, Power and Computing Technologies, ICCPCT 2016, 2016. doi: 10.1109/ICCPCT.2016.7530355. [3] Y. Liu and L. Luo, "Lossless compression of full-surface solar magnetic field image based on huffman coding," in Proceedings of the 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2017, 2017. doi: 10.1109/ITNEC.2017.8284866. [4] Z. Shao et al., "A High-Throughput VLSI Architecture Design of Canonical Huffman Encoder," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 69, no. 1, 2016, doi: 10.1109/TCSII.2016.3091611. [5] S. R. Khaitu and S. P. Panday, "Canonical Huffman Coding for Image Compression," in Proceedings on 2017 IEEE 3rd International Conference on Computing, Communication and Security, ICCCS 2017, 2017. doi:10.1109/CCCS.2017.8586816. [6] Xilinx, "Vivado HLS Optimization Methodology Guide 2017.1," Xilinx.Com, vol. 1270, 2017. [7] Xilinx, "Designing Protocol Processing Systems with Vivado High-Level Synthesis," Xapp1209, vol. 1209, 2014. [8] Xilinx, "UltraFast Design Methodology Guide for the Vivado Design Suite," Ug949, vol. 949, 2016. [9] A. Pal, Low-power VLSI circuits and systems. 2015. doi: 10.1007/978-81-322-1937-8. [10] J. Oh and M. Pedram, "Power reduction in microprocessor chips by gated clock routing," Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC, 1998, doi: 10.1109/aspdac.1998.669478. [11] J. Oh and M. Pedram, "Gated clock routing for low-power microprocessor design," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 20, no. 6, 2001, doi: 10.1109/43.924825.