International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp994-1003

High-Efficiency Multicontext MQ Arithmetic Coder Designed using VLSI design on FPGA

Mr.M.Ramana Reddy [(1)], Mr.B.Ajantha Reddy [(2)], Tetanela Gowtham Raj [(3),] Devaki Sai Akshay [(4),] Duggempudi Rama Krishna [(5)]
[1,2] *Krishna Chaithanya Institute Of Technology & Sciences, Ece Department, Markapur, Andhra Pradesh.*
[3,4,5,] *Krishna Chaithanya Institute Of Technology & Sciences, UG Student-ECE, Markapur, Andhra Pradesh.*

Abstract. In this Paper MQ arithmetic coding involves intrinsic serial processes, it is an adaptive arithmetic coding that evolved from Q coding and has become a significant throughput bottleneck of JPEG2000 compression. To address the bottleneck, the multicontext MQ coder's high-performance hardware architecture is suggested in this brief. Concurrent coding for two adjacent more likely symbols (MPSs) can be accomplished by the suggested architecture. The suggested coder achieves a throughput of 506.93 MSymbols/s under 0.5 bpp bit rate, according to performance analysis results, while consuming 1.61 CXD pairs each cycle. In addition to achieving high throughput, the suggested architecture also keeps power and hardware consumption low. In contrast to the cutting-edge two-context coder, resulting in a 38% increase in the figure of merit (FoM). There is a 49% reduction in the power-delay product (PDP) when compared to the single-context coder. The Project can be implemented and functionally synthesized by using Xilinx Vivado Verilog Hdls.

Keywords: MQ coder , FPGA, Verilog HDL,Jpeg2000 Vlsi Design.

I.INTRODUCTION

JPEG2000 is a popular still image compression standard, which provides excellent compression performance. High-speed hardware implementations of JPEG2000 are critical for many real-time applications such as digital photography, medical imaging, and satellite imaging. The most crucial algorithm adopted in JPEG2000 is the embedded block coding with optimal truncation (EBCOT). It is also the most computationally intensive component of JPEG2000. The EBCOT Tier-1 engine accounts for most of the processing time in JPEG2000. There are two processing stages in the EBCOT Tier-1 engine. The first stage is bit-plane coding (BPC) and the second one is arithmetic encoding (AE). The BPC processes wavelet coefficients of the codeblocks and generates context-decision (CXD) pairs. The AE is also known as the MQ coding since the coding algorithm is based on the Q coder [1]. The MQ coding processes the CXD pairs from the BPC and produces a compressed bitstream. It is now proved that multiple input samples of the BPC can be encoded in parallel with fewer hardware resources, so the coding efficiency of the BPC can be improved effectively [2]. However, the throughput of MQ coders can hardly keep up with that of the BPC, which makes the MQ coding a major throughput bottleneck of the JPEG2000 encoding system. Thus, high-performance architectures of the MQ coding must be designed carefully.

The MQ coding is serial in nature. Its processing speed is seriously affected by high dependence between the adjacent CXD

pairs. To achieve high throughput, some architectures are designed to improve the operating frequency [3], [4], [5]. Sarawadekar and Banerjee [3] proposed a three-stage, parallel-pipelined MQ coder, where the renormalization and byte-out stages work concurrently. In [4], an "index-arbiter" method is proposed to prevent the pipeline from stalling. In [5], the probability estimation representation is modified to minimize memory consumption. Despite high throughput, the coders mentioned above have a common drawback of processing only one CXD pair per clock cycle. So, these MQ coders must operate at high frequency to meet the throughput requirement, which leads to high power consumption. To reduce the power consumption and avoid the potential clock-domain crossing issues, several methods are proposed to design MQ coders that consume multiple CXD pairs in one cycle [6], [7], [8]. Zhou and Bao-Jun [6] proposed a twocontext architecture, in which the byte-out procedure is pipelined to reduce the critical path. In [7] and [8], the probability estimation table (PET) is expanded to store more parameters for two-context coding. Compared with single-context architectures, multicontext coders have lower power consumption. However, they require much more hardware resources to cope with the high dependence between two adjacent CXD pairs.

In general, the coders achieve high throughput, but they fail to keep the balance between hardware cost and power consumption. Given that the MQ coder is the basic unit of highspeed JPEG2000 encoders and is reused on a large scale, a practical MQ architecture should be designed to reach high throughput while achieving both low hardware utilization and low power consumption. To meet this requirement, in this brief, a multicontext hardware architecture of MQ coding is proposed based on a novel partially parallel coding strategy. The proposed architecture overcomes the disadvantages of existing single-context coders and multicontext coders, thus achieving both high area efficiency and high power efficiency.

2. OVERVIEW OF THE MQ ALGORITHM IN JPEG2000

The MQ coder is an adaptive binary arithmetic coder which encodes the decision bit (D) according to its context (CX). There are 19 contexts specified in JPEG2000. Each of these contexts has an associated state which consists of an index to the PET and the value of the more probable symbol (MPS). The context states are stored in a dynamic index lookup table (ILT). The decision bit, also known as a symbol, is either an MPS or a less probable symbol (LPS). As an example, Fig. 1 shows the MQ coding of four consecutive symbols from left to right. Types of symbols are marked at the bottom. When a symbol is coded, the probability interval will be subdivided into two subintervals. One corresponds with the MPS and the other corresponds with the LPS. The LPS interval has the length of Qe. The next probability interval will be selected from one of the subintervals corresponding to the type of D. Then the length of the interval (A) and the base of the interval (C) will be updated. As is shown in Fig. 1, the length between the horizontal thick lines represents A. The levels of the lower thick lines represent C. To calculate the new value of A and C, Qe is found in the PET. Specifically, a symbol updates A and C as follows.

International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp994-1003

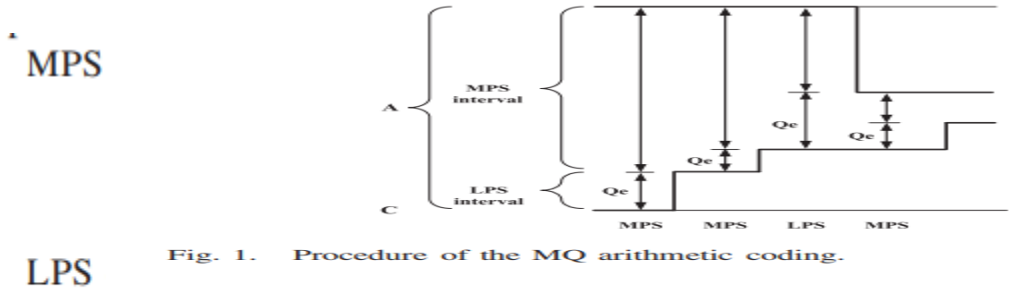

Fig. 1. Procedure of the MQ arithmetic coding.

In the calculation process, a 16-bit register A contains the value of the probability interval, and a 28-bit register C contains the lower bound of the interval. If the value of register A falls below 0x8000, which is equivalent to the decimal value of 0.75, both registers A and C are left shifted until A > 0x8000. Meanwhile, the index of the current context is updated. This procedure is called renormalization. When the number of shifts that occurred in registers A and C is enough, the upper bits of register C are emitted as the output byte that forms the final compressed bitstream. This procedure is called byte-out.

II. PROPOSED METHOD AND ITS METHODOLOGY

1.EXISTING SYSTEM: CODING STRATEGY

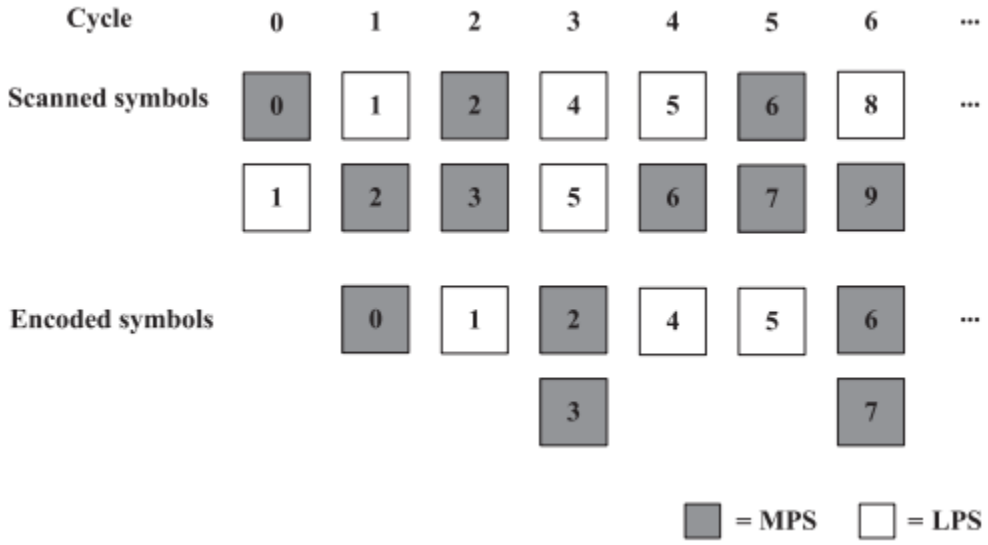

FIG.2. partially parallel coding strategy.

metacontext MQ coder is based on a partially parallel coding strategy shown in Fig. 2, where the scanned symbols and the encoded symbols in the first seven cycles are listed. A gray box represents an MPS and a white box represents an LPS. The number in the boxes represents the input sequence of the symbols. Two sequential

https://ijgst.com.2024.v13.i2.pp994-1003

input CXD pairs are scanned concurrently and the corresponding probability subintervals of each pair are examined. If both CXD pairs correspond with the MPS coding, they are processed in parallel in the next cycle. Otherwise, only the first CXD pair is encoded, while the second one is rescanned. In other words, the proposed coder can encode two adjacent MPS symbols (MPSMPS symbols) concurrently. The advantages of the proposed strategy are elaborated as follows.

There are four types of combinations of two CXD pairs. They are MPSMPS, MPSLPS, LPSMPS, and LPSLPS symbols. The MPSMPS symbols are encoded concurrently due to the following advantages. 1) Lower computational complexity. 2) High probability of occurrence. The complexity of the proposed coding strategy and the fully parallel strategy are compared in Table I. The lower complexity of the proposed strategy is due to fewer updated indices in renormalization and fewer emitted bytes in byte-out. When renormalization occurs, one CXD pair has two possible updated indexes: 1) one for the MPS and 2) one for the LPS. While an LPS must cause renormalization, an MPS may not cause it. As a result, there are six possible updated indices for the concurrent coding of all four combinations. However, the MPSMPS combinations have only two possible updated indices. So, the proposed strategy requires only three updated indices: two for MPSMPS symbols and one for LPS, which means the index update logic is simplified and much hardware utilization is saved in renormalization. On the other hand, an MPS causes no more than two times of left shift, while an LPS may cause a maximum of 15 times of left shift. Encoding of two CXD pairs may emit a maximum of four bytes (this occurs when the input combination is LPSLPS). Since the MPSMPS symbols emit no more than one byte, a large number of serial operations can be avoided and the critical path delay will not be established in byte-out.

2.PROPOSED SYSTEM: MULTICONTEXT MQ ARCHITECTURE

The overall architecture of the proposed MQ coder is shown in Fig. 3. The architecture has four pipeline stages. In the first stage, indices of the input CXD pairs are predicted and the subintervals are judged. In the second stage, the coder updates the value of register A. In the third stage, register C is updated. In the last stage, the byte-out procedure is performed. The signal num, which denotes the number of consumed CXD pairs, will feedback to the input queue.

International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp994-1003

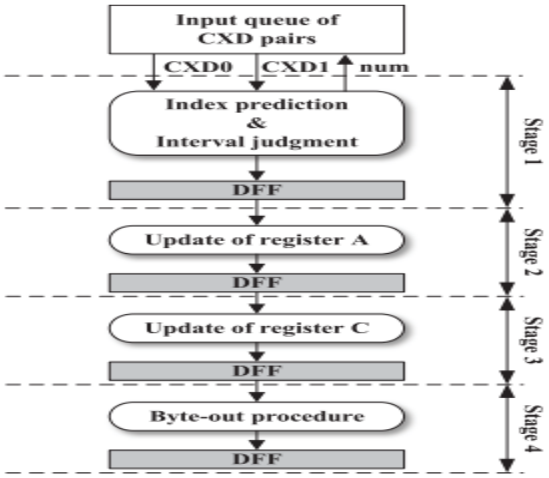

Fig. 3. Overall architecture of the proposed MQ coder.

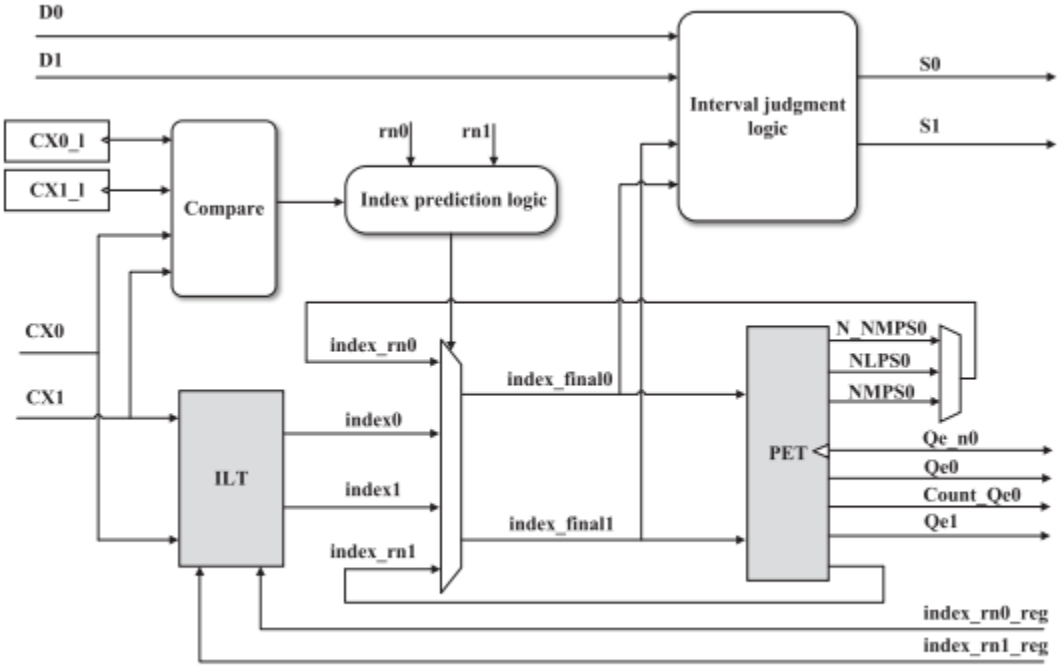

Fig. 4. Architecture of index prediction and interval judgment.

International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp994-1003

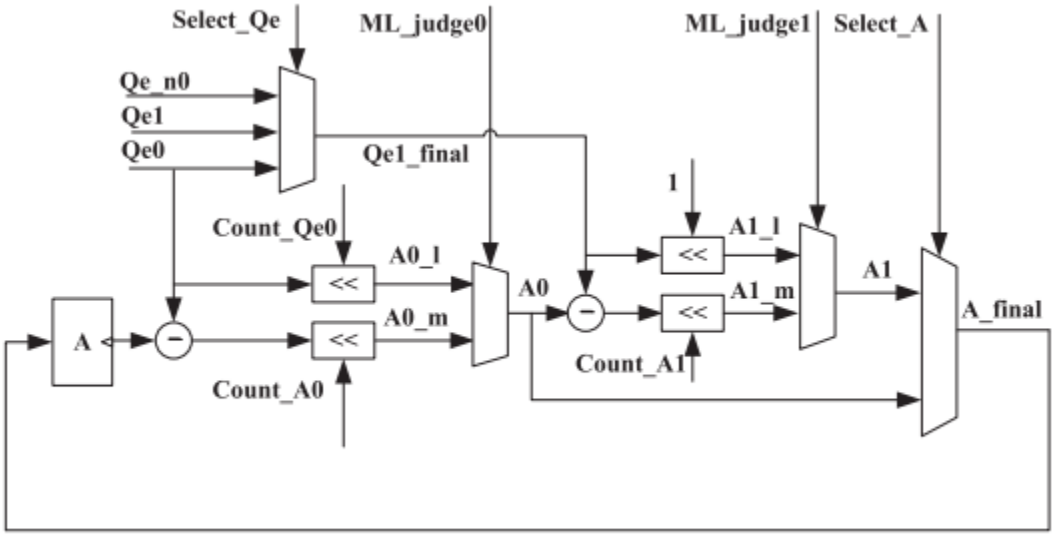

Fig. 5. Update of register *A*.

The proposed coder uses an extended PET to achieve the concurrent coding of MPSMPS symbols. There are 47 entries in the original PET and each entry contains four values. They are probability estimate for the LPS (Qe), the updated index of the MPS (NMPS), the updated index of the LPS (NLPS), and the probable symbol change of the MPS (SWITCH). As is shown in Table III, the extended PET stores three additional values: 1) N_NMPS; 2) Qe_n; and 3) Count_Qe. The value N_NMPS contains the updated index of the next MPS and Qe_n contains the probability estimation for the updated index of the MPS. Both of them are required to cope with the condition when two MPS symbols are encoded concurrently and the first one invokes renormalization. The value Count_Qe is the number of leading zero of Qe, which is used to simplify the shifting operations in renormalization. In addition, the value SWITCH is unrolled with the technique proposed in [9].

An efficient architecture of index prediction and interval judgment is proposed to handle concurrent coding. The architecture is shown in Fig. 4 and illustrated in Algorithm 1. In the proposed strategy, the encoded CXD pairs in each cycle are divided into three cases: two MPS symbols with different contexts, two MPS symbols with the same context, and a single CXD pair. Indices of both CXD pairs should be predicted in the first case, while only the index of the first CXD pair is required in the second and third cases. This is because the parameters of the second CX are stored in the extended PET. As is shown in Fig. 4, the predicted indices, index_ f inal0 and index_ f inal1, are selected between the original indices and the updated indices. If the same contexts encoded in the last cycle cause renormalization, updated indices, index_r n0 or index_r n1, will be chosen. Otherwise, the original indices, index0 or index1, will be chosen. Once the predicted indices are selected, Qe and other parameters are obtained from the PET for the update of register A and C. Meanwhile, the subintervals of each CXD pair, S0 and S1, are judged according to the values of input symbols D0 and D1. In the next cycle, the updated indices, index_r n0_reg and index_r

https://ijgst.com.2024.v13.i2.pp994-1003

n1_reg, are written into the ILT, if necessary.

Register A is updated as shown in Fig. 5. The architecture consists of two parts, and each part performs an update. The signals A0 and A1 are the results of the first and the second updates, respectively. The coder selects the result Affinal between A0 and A1 according to the number of consumed pairs. In the first update, values of Qe0 and Count_Qe0 are provided by the extended PET mentioned above. And Count_A0 denotes the number of leading zero of A − Qe0. The signals A0_m and A0_l are the results of (1) and (3) after renormalization, respectively. The correct results are then determined according to the corresponding subinterval M L_ judge0. The second update is like the first one. Significantly, the probability estimation of the second CXD pair Qe1_ final has three possible values. They are the probability estimation of the first pair (Qe0), the estimation of the updated index of the first pair (Qe_n0), and the estimation of a different context (Qe1).

III. RESULTS AND ANALYSIS DISCUSSION

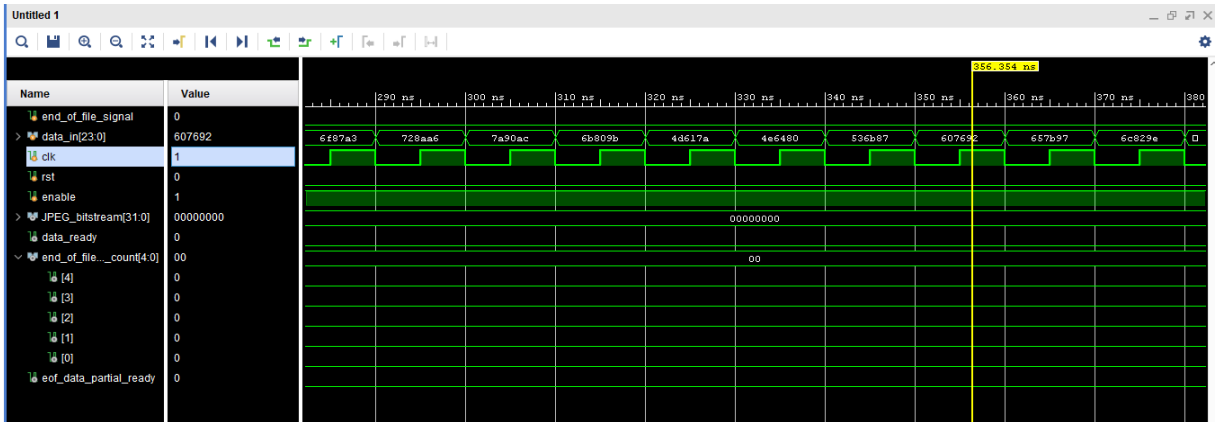

FIG1.SIMULATION OUTPUT

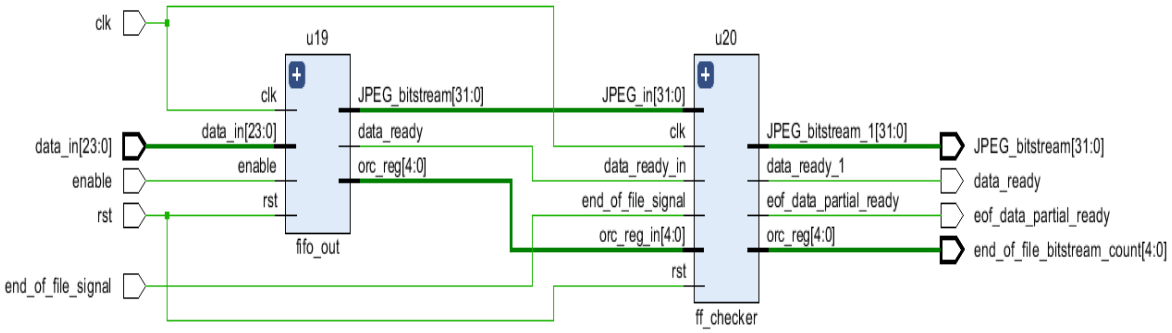

FIG2.RTL SCHEMATIC DIAGRAM

International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp994-1003

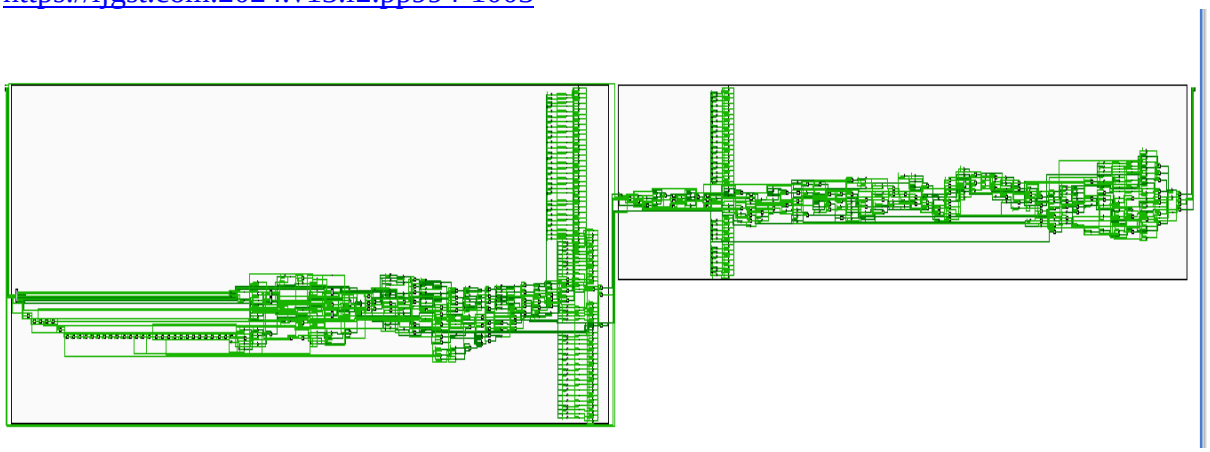

FIG3.RTL SCHEMATIC DIAGRAM 2.

In this brief, a high-performance multicontext architecture of MQ coding is proposed. A partially parallel coding strategy is applied to simplify the renormalization and the byte-out procedure in concurrent coding. A four-stage pipeline and an extended PET are designed to cope with the encoding of MPSMPS symbols. The coder is capable of consuming 1.61 CXD pairs per cycle on average and achieves up to 506.93 MSymbols/sec under 0.5 bpp. The proposed architecture can achieve high throughput while keeping a good balance between area efficiency and power efficiency. The proposed MQ coder presents a useful reference for the hardware implementation of a high-performance arithmetic coder in JPEG2000 and other image compression standards.

Arithmetic coding is a form of entropy encoding used in information theory and data compression. It is a variable-length coding technique that assigns codes to individual symbols and sequences based on their probability of occurrence. The term "MQ" refers to the implementation of arithmetic coding used in the JPEG2000 image compression standard, where "MQ" stands for the "modified quantization" algorithm used in conjunction with arithmetic coding.

Basic Steps of MQ Arithmetic Coding:

Initialization:

Define a probability model that assigns probabilities to different symbols based on their frequency of occurrence.

Interval Mapping:

Map each symbol to a corresponding interval on the unit interval (0, 1), with the size of each interval proportional to the symbol's probability.

Encoding:

Iterate through the input data symbol by symbol.

Narrow down the current interval based on the probability model for each symbol.

Output bits based on the narrowing of the interval.

Update the probability model based on the encoded symbols.

Decoding:

Initialize the same probability model used in encoding.

For each received bit, narrow down the interval based on the probability model.

Determine the symbol associated with the narrowed interval.

Update the probability model based on the decoded symbols.

https://ijgst.com.2024.v13.i2.pp994-1003

MQ in JPEG2000:
The Modified Quantization (MQ) algorithm is used in conjunction with arithmetic coding in the JPEG2000 image compression standard. In JPEG2000, MQ arithmetic coding is applied to the quantized wavelet coefficients.

MQ Coder:
The MQ coder is a finite-state machine used for entropy coding. It operates on the context-based probability models derived from the quantized wavelet coefficients.

Interval Mapping in MQ Coder:
The MQ coder maps a set of probability states to intervals on the unit interval (0, 1). The narrower the interval, the more bits are emitted during encoding and consumed during decoding.

Significance Propagation:
The MQ coder employs significance propagation to efficiently encode the significance and refinement bitplanes of the quantized wavelet coefficients.

Termination and Flushing:
The encoding process is terminated by flushing any remaining bits in the MQ coder's internal state to the output bitstream.

CONCLUSION

This paper, High-performance multicontext architecture of MQ coding is proposed. A partially parallel coding strategy is applied to simplify the renormalization and the byte-out procedure in concurrent coding. A four-stage pipeline and an extended PET are designed to cope with the encoding of MPSMPS symbols. The coder is capable of consuming 1.61 CXD pairs per cycle on average and achieves up to 506.93 MSymbols/sec under 0.5 bpp. The proposed architecture can achieve high throughput while keeping a good balance between area efficiency and power efficiency. The proposed MQ coder presents a useful reference for the hardware implementation of a high-performance arithmetic coder in JPEG2000 and other image compression standards.

3.REFERENCES

[1] W. B. Pennebaker, J. L. Mitchell, G. G. Langdon, and R. B. Arps, “An overview of the basic principles of the Q-coder adaptive binary arithmetic coder,” IBM J. Res. Develop., vol. 32, no. 6, pp. 717–726, Nov. 1988.
[2] R. Ghodhbani, T. Saidani, L. Horrigue, and M. Atri, “An efficient pass-parallel architecture for embedded block coder in JPEG 2000,” J. Real-Time Image Process., vol. 16, no. 5, pp. 1595–1606, Oct. 2019.
[3] K. Sarawadekar and S. Banerjee, “An efficient pass-parallel architecture for embedded block coder in JPEG 2000,” IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 6, pp. 825–836, Jun. 2011.
[4] Z. Di, Y. Hao, J. Shi, and P. Ma, “A high-throughput VLSI architecture design of arithmetic encoder in JPEG2000,” J. Signal Process. Syst., vol. 81, no. 2, pp. 227–247, Nov. 2015.
[5] T. Salem and H. Mahmoud, “Design of a high speed architecture of MQ-coder for JPEG2000 on FPGA,” Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 6, pp. 1–8, 2017.
[6] P. Zhou and Z. Bao-Jun, “High-throughout hardware architecture of MQ arithmetic coder,” in Proc. IEEE 10th Int. Conf. SIGNAL Process., Oct. 2010, pp. 430–433.

https://ijgst.com.2024.v13.i2.pp994-1003

[7] N. R. Kumar, W. Xiang, and Y. Wang, "Two-symbol FPGA architecture for fast arithmetic encoding in JPEG 2000," J. Signal Process. Syst., vol. 69, no. 2, pp. 213–224, Nov. 2012.
[8] M. Dyer, D. Taubman, S. Nooshabadi, and A. K. Gupta, "Concurrency techniques for arithmetic coding in JPEG2000," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 53, no. 6, pp. 1203–1213, Jun. 2006.
[9] N. Bao, Z. Jiang, Z. Qi, and W. Zhang, "High-throughput MQ encoder for pass-parallel EBCOT in JPEG2000," in Proc. 28th IEEE Int. System Chip Conf. (SOCC), Sep. 2015, pp. 410–414.