Shashank Srivastava, 2021, 9:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

Efficient Hardware Implementation of Extended Go lay Code for Error Correction Parallel Decoder-DSP Application

M. Tech. Scholar Shashank Srivastava, Associate Prof. Dr Anshuj Jain
Department of Electronics & Communication Engineering,
SCOPE College of Engineering,
Bhopal, India

Abstract- The Channel coding is commonly incorporated to obtain sufficient reception quality in advance wireless mobile communications transceiver to counter channel degradation due to inter-symbol interference, multipath dispersion High speed and high throughput hardware for encoder and decoder could be useful in communication field. Due to the channel achieving property, the GOLAY code has become one of the most favorable error-correcting codes. This paper presents VLSI Implementation of Extended Go lay Code for Error Correcting Parallel Decoder FPGA-DSP Applications, which outperform the existing architectures in terms of speed and throughput. The Simulation is done using the Xilinx ISE 14.7 platform with Ishim test.

Keywords: Go lay, Error, FPGA, DSP, VLSI, Correcting, Code, Wireless, Transceiver.

I. INTRODUCTION

The Go lay code was utilized to provide error control on the voyager mission. An algebraic decoding algorithm for the Go lay code is given to correct the three possible errors. The shift-search procedure compares favorably in complexity and speed with the completely Elia decoding method. The algebraic technique is slightly faster than that of the shift-search procedure. In this work, based on the idea of, a novel reduced lookup table method is developed to decode the (23, 12, 7) Go lay code.

The reduced lookup table using in this algorithm consists of syndrome patterns and corresponding error patterns which only have one and two errors in the message block of the codeword. The proposed method works as follows: Given a received codeword r, first, the syndrome s is computed and then the weight of this syndrome w(s) is computed directly.

If w(s) =0, it means no errors happened in the received codeword. If w(s) ≦3, it means at most three errors happened in the parity check block of the received codeword. One just need to do is shifting the syndrome left k bits to form a 23-bit length word, and then the received codeword minus (modulo 2) this 23-bit length word to correct the received codeword If w(s) ≧4, it means at least one error happened in the message block of the received codeword.

First, one searches if this syndrome matches the syndrome pattern listed in the reduced lookup table. If the syndrome is in the table, it means at most 2 errors happened in the message block of the received codeword, and then the received codeword minus (modulo 2) the error pattern corresponding to the syndrome to correct the received codeword. Second, if the syndrome is not in the table, it means there are three possible conditions: 1. one error in the parity check block two errors ISSN NO: 0898-3577 Page No: 8 Compliance Engineering Journal Volume 12, Issue 10, 2021 in the message block, 2. two errors in the parity check block one error in the message block, or 3. three errors in the message block.

For 1st condition, using syndrome minus (modulo 2) syndrome pattern in the table to obtain the difference and compute the weight of this difference respectively. If this weight equals to 1, it means there is one error in the parity check bit. So, shifting the difference left k bits to form a 23-bit length word,

International Journal of Science, Engineering and Technology

An Open Access Journal

and then the received codeword minus (modulo 2) this 23-bit length word and minus corresponding error pattern to correct the received codeword. For the 2nd and 3rd conditions, the received codeword must cyclic shift left n-k bits, and therefore parity check block and message block are swapped, which indicates that the errors are also swapped. Repeating the procedures above, one can correct up to three errors in the received codeword.

II. METHODOLOGY

The coding scheme is developed for the data encoding and decoding. Many of the codes like block code, cycle codes are already developed and research is continue going on the enhancement of the code. The extended binary Go lay code, G24 encodes 12 bits of data in a 24-bit word in such a way that any 3-bit errors can be corrected or any 7-bit errors can be detected.

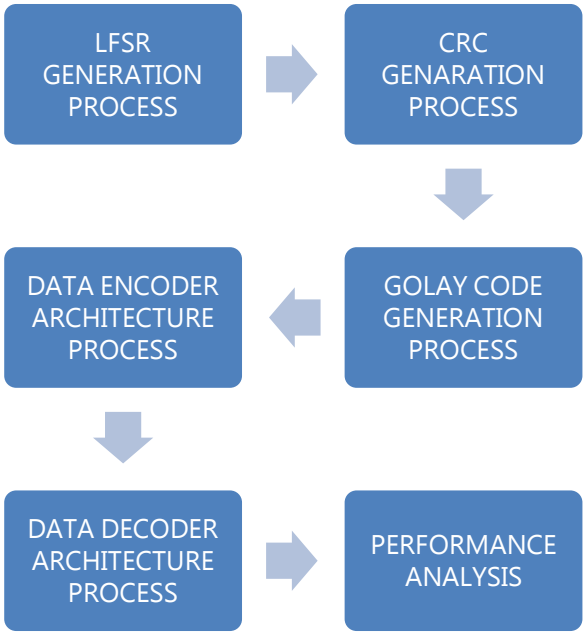

Fig 1. Flow Chart.

Design a secure encoder and decoder architecture using GOLAY code function. So we create the key generation process based on polynomial equation. This equation is present in GALOIS field process and to update the key value in every input data bits.

This process is used to improve the data secure transmission process. The pseudo random pattern generation process is mainly used to generate the pattern results based on normalized distance. This work is to modify the division operation architecture in encode and decode function. We apply the xor gate operation in subtraction process and to design a priority based encoder design. This design is to analysis the subtraction data bits and to add the no of '0' bits. This process is to reduce the overall division architecture level. GOLAY code function is to add the addition key data in encoder operation. This process is to modify the data and key addition process. This function in mainly based on majority architecture design process.

This work is to increase the secure level and to optimize the circuit complexity. Proposed system is to modify the encoder and decoder data bits structure level and to add the message bit, key bit and to apply the these bits into GOLAY binary code technique. This technique is to apply the majority gate analysis process and to get the final majority output bit and to add the any location in encoder architecture output data bits.

To combat this problem, a hardware module programmed to yield a Go lay encoded codeword may be used. Go lay decoder is used extensively in communication links for forward error correction. Therefore, a high speed and high throughput hardware for decoder could be useful in communication links for forward error correction.

III. SIMULATION RESULTS

The implementation of the proposed algorithm is done over Xilinx ISE 14.7. The ISE package processing toolbox helps us to use the functions available in Xilinx Library.

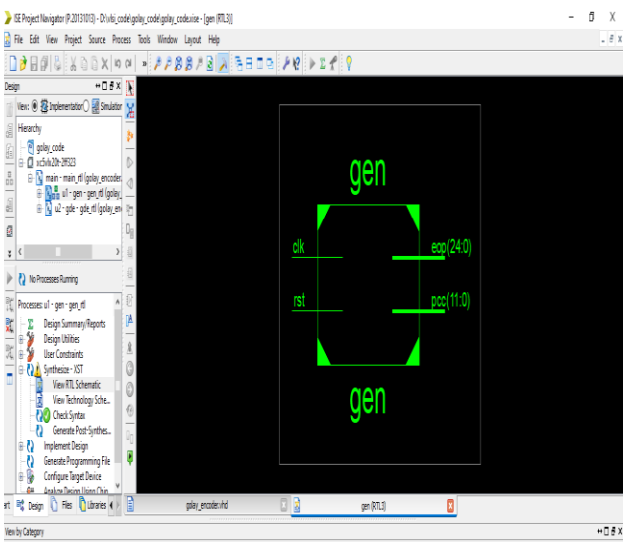

Fig 2. RTL View of Top module.

International Journal of Science, Engineering and Technology

An Open Access Journal

Figure 2 presents the top level view of the Register transfer level. Initially clock pulse set in logic 1 and applies reset also high using 1. Output shows the 24 bit golay code and 12 bit original message.

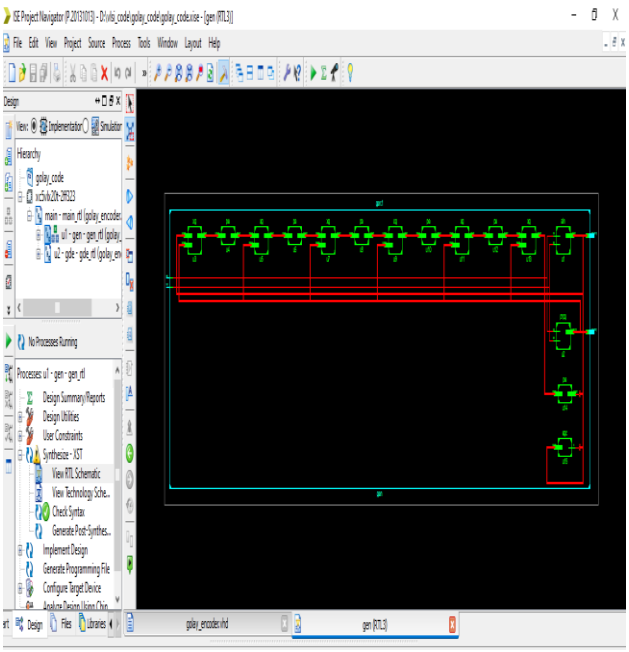

Fig 3. Complete RTL View.

Figure 3 shows the complete RTL view of the go lay code. It also shows the technological view as select during the top module processing.

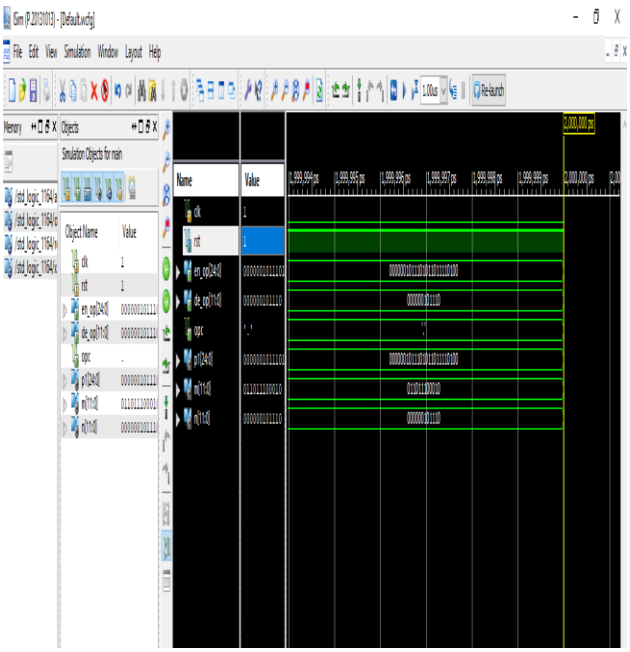

Fig 4. Test bench results-Data input.

Figure 4 provides the test bench results. Here clock is set at logic 1 or active condition, reset is at 1. Then run the current signal status. The go lay code value is 000000101110101101111010 0. Data value is 0000000101110. Opc is in null status.

Fig 5. Test bench results-Retrieving Stage.

Figure 5 shows the running status of the golay code. The data is starting to retrieve as reset set on the 0 signal. The opc set on the error state and data is retrieving.

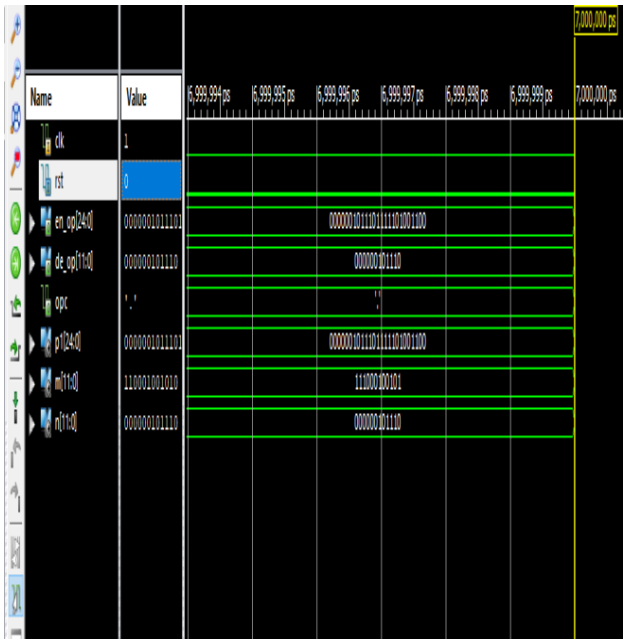

Fig 6. Test bench results- Optimized results.

Figure 6 shows the running status of the go lay code. The data is retrieved and opc is set at initial stage. The go lay code provides the original data during the transmission at very less stage. Here the optimized results show the go lay (24, 12, 6).

International Journal of Science, Engineering and Technology

An Open Access Journal

Table 1. Comparison of simulation results.

Sr No.	Parameters	Previous Result [1][8]	Proposed Result	Improvement in (%)
1	Methodology	Golay code (24,12,5), (24,12,8)	Golay code (24,12,3) (24,12,6)	NA
2	Area	493	318 (6.6 %)	Aprox 35%
3	Decoder Delay	3.11 ns	1.599ns	Aprox 50%
4	Power	0.76 mw	0.45 mw	Aprox 30%
5	Frequency	NA	625.332 MHz	NA
6	Throughput	NA	7×10^{9} or 7GHz	NA

IV. CONCLUSION

This paper presents VLSI Implementation of Extended Go lay Code for Error Correcting Parallel Decoder FPGA-DSP Applications, which outperform the existing architectures in terms of speed and throughput.

The proposed architectures were simulated and tested on Virtex-5 platform. Although the CRC encoder and decoder is intuitive and easy to implement, and to reduce the huge hardware complexity required. The proposed go lay code gives the better performance in terms of the calculated parameters. The proposed go lay code optimized the (24, 12, 3) to (24, 12, 6) level. The optimized area or component is 318 (6.6 %) while previously it was 493. The delay or latency value is 1.599 ns while it was 3.11 ns in existing work. The optimized power is 0.45 mw while previous it is 0.76 mw. Proposed technique is to reduce the circuit complexity for data transmission and reception process.

REFERENCES

[1] P. J. Edavoor, S. Raveendran and A. D. Rahulkar, "Approximate Multiplier Design Using Novel Dual-Stage 4:2 Compressors," in IEEE Access, vol. 8, pp. 48337-48351, 2020, doi: 10.1109/ACCESS .2020.2978773.

[2] F. Sabetzadeh, M. H. Moaiyeri and M. Ahmadinejad, "A Majority-Based Imprecise Multiplier for Ultra-Efficient Approximate Image Multiplication," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 66, no. 11, pp. 4200- 4208, Nov. 2019.

[3] H. Saadat, H. Bokhari and S. Parameswaran, "Minimally Biased Multipliers for Approximate Integer and Floating-Point Multiplication," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 11, pp. 2623- 2635, Nov. 2018.

[4] V. Mrazek, Z. Vasicek, L. Sekanina, H. Jiang and J. Han, "Scalable Construction of Approximate Multipliers With Formally Guaranteed Worst Case Error," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 11, pp. 2572-2576, Nov. 2018.

[5] S. Venkatachalam and S. Ko, "Design of Power and Area Efficient Approximate Multipliers," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 5, pp. 1782-1786, May 2017.

[6] S. Mazahir, O. Hasan, R. Hafiz and M. Shafique, "Probabilistic Error Analysis of Approximate Recursive Multipliers," in IEEE Transactions on Computers, vol. 66, no. 11, pp. 1982-1990, 1 Nov. 2017.

[7] W. Liu, L. Qian, C. Wang, H. Jiang, J. Han and F. Lombardi, "Design of Approximate Radix-4 Booth Multipliers for Error-Tolerant Computing," in IEEE Transactions on Computers, vol. 66, no. 8, pp. 1435-1441, 1 Aug. 2017.

[8] A. Mokhtari, W. Shi, Q. Ling and A. Ribeiro, "DQM: Decentralized Quadratically Approximated Alternating Direction Method of Multipliers," in IEEE Transactions on Signal

Processing, vol. 64, no. 19, pp. 5158-5173, 1 Oct.1, 2016.

[9] H. Jiang, J. Han, F. Qiao and F. Lombardi, "Approximate Radix-8 Booth Multipliers for Low-Power and High-Performance Operation," in IEEE Transactions on Computers, vol. 65, no. 8, pp. 2638-2644, 1 Aug. 2016.

[10] B. Shao and P. Li, "Array-Based Approximate Arithmetic Computing: A General Model and Applications to Multiplier and Squarer Design," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 62, no. 4, pp. 1081-1090, April 2015.

[11] Z. Zhi-Qing Lü and X. An, "Non-conforming finite element tearing and interconnecting method with one Lagrange multiplier for solving large-scale electromagnetic problems," in IET Microwaves, Antennas & Propagation, vol. 8, no. 10, pp. 730-735, 15 July 2014.

[12] D. De Caro, N. Petra, A. G. M. Strollo, F. Tessitore and E. Napoli, "Fixed-Width Multipliers and Multipliers-Accumulators With Min-Max Approximation Error," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 60, no. 9, pp. 2375-2388, Sept. 2013.

[13] J. Wang, S. Kuang and S. Liang, "High-Accuracy Fixed-Width Modified Booth Multipliers for Lossy Applications," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 19, no. 1, pp. 52-60, Jan. 2011.

[14] F. Auger, Z. Lou, B. Feuvrie and F. Li, "Multiplier-Free Divide, Square Root, and Log Algorithms [DSP Tips and Tricks]," in IEEE Signal Processing Magazine, vol. 28, no. 4, pp. 122-126, July 2011.

[15] Park and T. Kim, "Multiplier-less and table-less linear approximation for square and square-root," 2009 IEEE International Conference on Computer Design, Lake Tahoe, CA, 2009, pp. 378-383.