Biosc.Biotech.Res.Comm. Special Issue Vol 13 No 13 (2020) Pp-177-184

BBRC
Bioscience Biotechnology Research Communications

Low Power VLSI Implementation of Convolution Encoder and Viterbi Decoder using Verilog HDL

Dasari Ramanna* and V Ganesan
[1]*Sathyabama institute of science and Technology, India*

ABSTRACT

Viterbi decoder is considered as one of the widespread error correcting channel decoder of communication devices. The Viterbi decoder is mainly used for decoding the convolution codes. The large amount of trellis transitions increases the computational complexity of the Viterbi decoder. In this paper, the finite state machine based trellis encoding is proposed in the convolutional encoder for minimizing the system complexity. Moreover, the folding transformation is proposed in the Viterbi decoder for minimizing the number of stages processed while decoding the encoded data bits. The main objective of this proposed system is that the reconstruction of original data bits with less errors. The performance of the proposed system is analyzed in terms of number of slice LUT, number of slice register, BRAM, delay and failure rate. In addition, the performance of the proposed method is evaluated with the duplication with comparison based protection method. The amount of slice registers utilized in the proposed system are 739 for Virtex 5, it is less when compared to the duplication with comparison based protection method.

KEY WORDS: CONVOLUTION ENCODER, ERRORS, FINITE STATE MACHINE BASED TRELLIS ENCODING, FOLDING TRANSFORMATION, VITERBI DECODER.

INTRODUCTION

In communication system, the data is transmitted from the source to the destination over the channel or medium (Subramani et al. 2020). The channels of the wireless communication are become noisy due to the deficiencies created by the multipath effect and channel distortions. Since, the accuracy of the received data is mainly based on the external noise and channel. This external noise generates the interferences to the signal and creates the errors in the transmitted data. The energy efficiency and bandwidth of the wireless communication system is improved by using the Error Correction Codes (ECC) (Sun, Y. and Cavallaro 2011) (Reddy, B.S. and Rao), (Bahrami and Vasic 2019). The Forward Error Correction (FEC) of channel coding is used in ECC for improving the communication channel's reliability (Mathana et al. 2013). The family of FEC codes is generally attractive for mobile communication systems. This FEC codes are considered as a part of channel coding standards for 3GPP-LTE (4th generation system) or CDMA2000 and UMTS (3rd generation system). The decoding algorithm (e.g. turbo decoding) improves the operational complexity which is used to obtain the coding gain. The energy consumption is considered as equal constraint to the throughput in communication process (Liu et al. 2012).

In present digital communication system, a variety of Error Correction Codes (ECC) are developed based on the channel noise (Prakash and Muthamizhse 2016). Additionally, some of the ECC standards used in the existing methods are given as follows: Hamming codes and convolution codes (Marriwala et al. 2013), turbo decoding (El Chall et al. 2015), Polar Codes (Leroux et al. 2012), soft-decision-based error correction (Kim and Sung 2012), low complexity chase decoding (Garcia-Herrero et al. 2011), unary error correction code (Zhang et al. 2014), etc.

ARTICLE INFORMATION

*Corresponding Author: ram413vlsi@gmail.com
Received 7th Oct 2020 Accepted after revision 23rd Dec 2020
Print ISSN: 0974-6455 Online ISSN: 2321-4007 CODEN: BBRCBA

Thomson Reuters ISI Web of Science Clarivate Analytics USA and Crossref Indexed Journal

Clarivate Analytics
crossref

NAAS Journal Score 2020 (4.31)

Online Contents Available at: http//www.bbrc.in/
Doi: http://dx.doi.org/10.21786/bbrc/13.13/25

The digital communication system is designed and implemented by using the Application-Specific Integrated Circuit (ASIC) (Cai et al. 2017). Generally, the convolution coding is used in the communication system's transmitter while transferring the data packets through noisy channel and convolutional decoding is utilized for retrieving the input signal data. In receiver section, the error is detected and corrected by adding the redundancy to a message using the convolution coding. However, the decoding consumes more time in the receiver baseband with the fast Fourier transform (Ates et al. 2016). The Viterbi algorithm is the common standard that is used instead of convolutional decoding to decode the convolution code. The computational complexity of Viterbi algorithm is exponentially increased with the higher constraint length (Vaithiyanathan 2015). The main contribution of the paper is given as follows:

- The Finite State Machine (FSM) based trellis encoding is used in the traditional convolution encoder to decrease the system complexity.
- Then the Viterbi decoder is integrated with the folding transformation to reduce the number of stages processed while decoding the encoded data bits.
- The proposed architectures of convolution encoder and Viterbi decoder are used to obtain lesser hardware utilization along with less complexity.

The overall organization of the paper is given as follows: the literature survey about the Viterbi decoder is given in the section 2. The proposed convolutional encoder and Viterbi decoder with FSM based trellis and folding transformation is described in section 3. The experimental and comparative analysis of the proposed system are given in section 4. Finally, the conclusion is made in section 5.

Literature Survey: Kermani et al. 2016 presented the error detection methods for the Carry Select Add (CSA) unit and Precomputed CSA (PCSA) structures. In addition, theCSA and PCSA are the structures of low-complexity and lowlatency Viterbi decoder. The developed architecture is mainly based on the fine-tuned recomputing with rotated operands as well as it is used to detect the permanent and transient faults which are coupled with signature based schemes. Therefore, the developed Viterbi decoder obtains higher error coverage as well as the performances of the decoder are improved in the communication systems. However, the erroneous result is obtained as output, when the single stuck is occurred in the CSA and PCSA.

Kuang 2018 developed the codeword based State Transparent Convolutional (STC) decoder. This developed STC contains the codeword-based error detector (CB-ED), codeword-based error corrector (CB-EC) and codeword-based Viterbi decoders (CB-VD). The erroneous of the collected codeword is identified by using the CB-ED and the CB-VD is utilized to correct the few corrupted codewords. Additionally, the affected codewords are identified and recovered by using the CB-EC. An active Register Exchange (RE) stage of Viterbi decoder with a clock-gating method is dynamically adjusted by using CB-VD. This RE dynamic adjustment is used for minimizing the switching activities. The low power decoder used in the codeword based STC decoder was required an additional space that increased its hardware utilization than the traditional architectures.

Xie et al. 2018 presented the Standard Convolutional Symbols Generator (SCSG) in multi-parameter reconfigurable Viterbi decoder. The SCSG block in Viterbi decoder is helps to adapt the multiple parameters and to minimize the resource consumption. Then the iterative approach is used in the SCSG to generate the standard convolutional symbols the all odd and even states. Additionally, the generated convolutional symbols are distributed to the branch metric (BM) block for calculating the branch metrics. The developed SCSG based Viterbi decoder minimizes the resource utilization which is used for BM calculation. The errors in the results are high due to the less constraint length of the Viterbi decoder.

Broich 2017 developed the guideline to design the turbo and Viterbi decoder data path with less widths. This is obtained by deriving the metric differences of the maximum absolute values of internal Max-Log-MAP decoder. Moreover, the maximum absolute values of difference are formulated for the State Metrics (SM) and Candidate SMs (CSM). This SM and CSM are the intermediate outputs of add-compare-select (ACS) operations and path metrics. The realization of energy and area was expensive due to the higher acquisition length of the input. Limitation.

Gao 2019 developed the Viterbi decoder in the Static Random Access Memory based FPGAs (SRAM-FPGAs). This work analyzed the soft error effects in the configuration memory of the Viterbi decoder. The Viterbi decoder is protected against from the errors by using Algorithm-Based Fault Tolerance (ABFT) protection method which depends on the structure of enhanced Duplication with Comparison (DWC). The developed fault tolerant architecture avoids the errors only from propagating to the output, when the Single Event Upset (SEU) is occurred in the SRAM-FPGAs. The DWC protected vitebi decoder creates an additional delay than the unprotected decoder.

Proposed System: In the proposed system, an effective regeneration of input data is obtained from the erroneous bits. The proposed system mainly comprises of three phases such as convolutional encoder, data bits transmission and Viterbi decoder. The Viterbi decoder is combined with the folding transformation for minimizing the number of stages while performing the decoding process. Additionally, the complexity of the convolutional encoder is reduced by using the FSM based trellis in the transmitter. The overall architecture of proposed system is shown in the Figure 1.

Figure 1: Block diagram of the proposed system

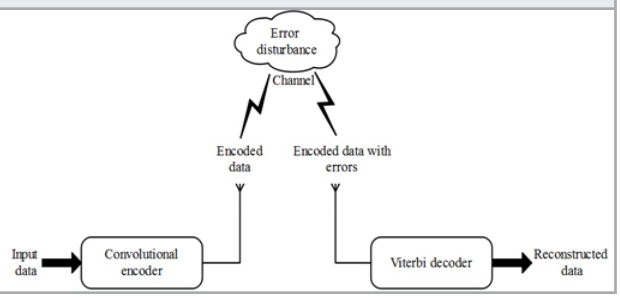


Table 1. Parameters of convolutional encoder

Parameters	Value
Constraint length (H)	4
Input data (I)	1
Encoded data (c)	2

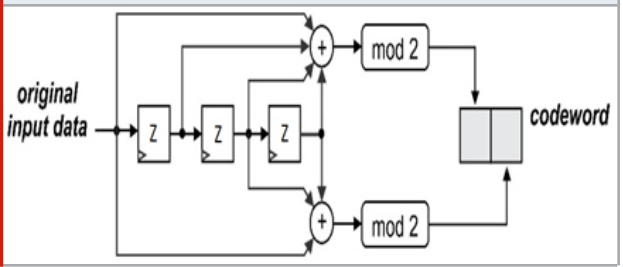
1. The convolution encoder generates the encoded data which specifies the input data and its redundant bits.
2. The encoded data is transmitted from the transmitter to the receiver through the transmission channel.
3. In the transmission channel, the encoded data may affect because of error disturbance caused from the environment.
4. The received data is affected by the errors and these erroneous data is required to be processed in the Viterbi decoder to reconstruct the original data which is given as input.

The architecture of convolutional encoder and Viterbi decoder used in this proposed system are clearly described as follows,

Convolutional encoder: In the proposed system, the convolution encoder is the 1st element of the convolutional error correction method. This convolutional encoder encodes the input data (I) into codewords(c). The important terms used for the computation of convolutional encoder are constraint length (H), number of states (N), code rate (R), input data (I) and codeword/ encoded data (c). The length of convolutional encoder process is represented by the constraint length. The amount of shift registers utilized in the procedure defines the H. The architecture of the convolutional encoder is shown in the Figure 2.

Equation (1) and (2) shows the expression of the polynomial codeword generator. The placement of XOR gates in the convolutional encoder is defined by the generator polynomial. The operation of input data and constraint length are defined the amount of states which is defined in the equation (3). The ratio among the amount of input data bits to the encoded data bit is defined as code rate (R) that is shown in equation (4). Additionally, the parameters considered in the convolutional encoder are represented in Table.

Figure 2: Architecture of Convolutional encoder



$$G1 = 1 + Z + Z^2 + Z^3 \quad (1)$$

$$G2 = 1 + Z^2 + Z^3 \quad (2)$$

Where, the delay operator is represented as Z.

$$N = 2^{\{I_{bitwidth}(H-1)\}} \quad (3)$$

$$R = \frac{I_{bitwidth}}{c_{bitwidth}} \quad (4)$$

Figure 3: FSM of convolutional encoder

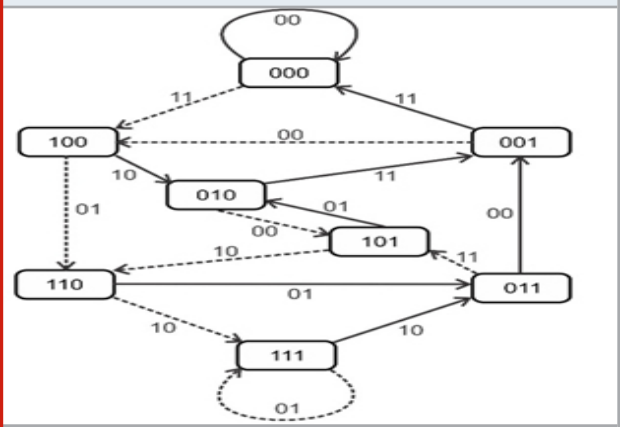
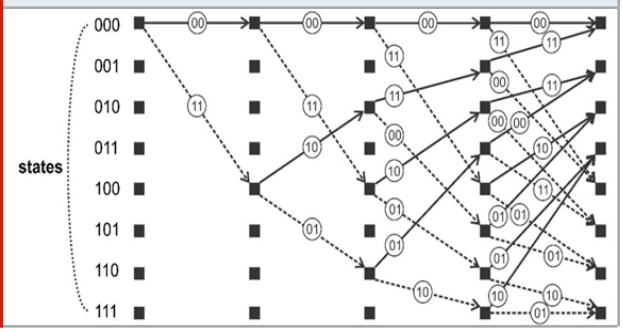


Figure 4: Trellis diagram

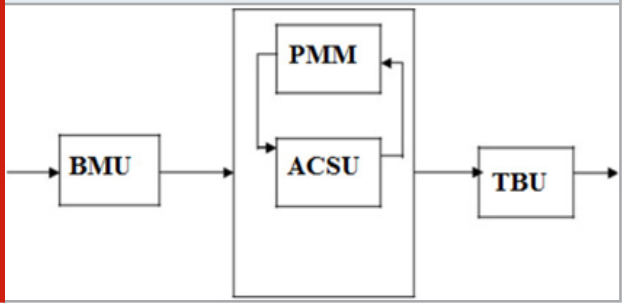


FSM based trellis diagram: The convolutional codes are described as trellis diagram for describing the transitions among states as a function of time. The FSM and the trellis network are represented in the Figure 3 and Figure 4 respectively. This represents the state values (N), predictors(p) and comparators (o). The 3-bit binary value specifies the state values that shows the convolutional

encoder totally has 8 states. In Figure 3, the solid arrow line specifies the value p=0 and dashed arrow line specifies the value p=1. Additionally, the comparator values are mentioned near to the arrow line.

Viterbi decoder: The encoded bit from the convolutional encoder is decoded by using the Viterbi algorithm in the Viterbi decoder. The path which enters each state in the trellis diagram is recursively determined by the Viterbi decoder. The sequence of codewords with less distance to the collected noisy codewords are defined the path of the state. Additionally, the state's Path Metric (PM) is the distance between the path and the collected codewords. The Branch Metric (BM) is added to the PM in the preceding time stage for determining the possible path for a state. Subsequently, it chooses the closest path as the possible path. The block diagram of the Viterbi decoder used in the proposed system is shown in the Figure 5. The basic functional blocks used in the decoder are Branch Metric Unit (BMU), Add–Compare–Select Unit (ACSU), Path Metric Memory (PMM), and Trace Back Unit (TBU).

Figure 5: Block diagram of the Viterbi decoder



Branch metric unit: The BMU computes the branch metric based on the Euclidean distance or hamming distance. The Euclidean distance and hamming distance are calculated by using the soft decision and hard decision viterbi decoder respectively. The operation performed in the BMU are comparison of collected code symbol with the probable code symbol and identification of amount of varied bits.

Add compare Select unit: The new path metric is obtained by adding the BM to the partial PM. The new PM is compared, when the two paths are entered into the same state. Here, the path which has the minimum metric is selected as the new path named as survivor path as well as the selection is accomplished for the all states. The selected PMs are stored in the Path Metric Memory (PMM). Subsequently, the survivor path's PM of an each state is updated and stored in the PMM.

Trace back unit: The output data and survivor path are determined in the final unit named as trace-back process or register exchange method. This is used for generating the decoding the output sequence.

Folding transformation: The folding transformation is developed in the Viterbi decoder for minimizing the number of stages processed while decoding the encoded data. The folding is a technique which is used to systematically determine the control circuits in architectures. In folding transformation, the multiple algorithm operations are time-multiplexed to a single hardware module. For example, the addition operations are combined into a pipelined ripple-carry adder. The folding transformation is identical to the loop folding that is used for high-level synthesis. The folding algorithm used in the Viterbi decoder is single-rate algorithms.

Consider an arc connecting nodes X and Y are connected with i delays in single-rate algorithms. The lth iteration of the nodes X and Y are scheduled to execute in the time units M_X l+x and M_Y l+y respectively, where the folding orders of X and Y are represented as x and y. The J_X and J_Y denotes the hardware operators (or functional units) which used for executing the node X and Y. The output of lth iteration of node X is available in M_X l+x+P_X, when the H_X is pipelined by the P_X stages. The output of X is utilized by the (l+i)th iteration of Y, when the arc X→Y contains i delays. Equation (5) specifies the amount of storage units used in the system.

$$D(X \to Y) = M_Y(l+i) + y - (M_X l + P_X + x)$$
$$= (M_Y - M_X)l + M_Y i - P_X + y - x \quad (5)$$

Where, i is the delay. The M_X is equal to the M_Y (i.e., M_X=M_Y=M), when the amount of delays to be independent of the iteration (l). Then the amount of required storage units for the folding transformation is expressed in the equation (6).

$$D(X \to Y) = Mi - P_X + y - x \quad (6)$$

In the proposed system, the data bits are transferred from the source to the destination is affected by the error. This is overcome by using the convolutional encoder in the source and Viterbi decoder in the destination. In that, the complexity of the convolutional encoder is minimized by using the FSM based trellis encoding. Additionally, the folding transformation is combined with the Viterbi decoder for minimizing the amount of stages processed in decoding process. This helps to minimize the hardware utilization and it has higher error coverage in the decoding process.

RESULTS AND DISCUSSION

The results and discussion of the proposed system is described in this section. The design and simulation of the proposed communication system is carried out in Xilinx ISE 14.2 software which is operated in a Windows 8 operating system with Intel core i3 processor and 4GB RAM. In this proposed system, the convolutional encoder module is developed with the FSM based trellis for encoding the input data bits. The constraint length of the convolutional encoder is 4 as well as the input and encoded data bits of the encoder is 1 and 2 respectively. Then the Viterbi decoder is combined with the folding transformation for decoding the encoded data bits. This

helps to reconstruct the input data which is transferred over the communication channel.

Performance analysis of the proposed system with different FPGA devices: The performance analysis of the proposed system with the convolutional encoder and Viterbi decoder is presented in this section. The proposed system is analyzed for three different FPGA devices such as Virtex 4, Virtex 5 and Spartan 6. The performances are analyzed in terms of slice LUT, slice registers, BRAM, delay and failure rate.

Table 2. Hardware analysis of proposed system for Virtex 4 device

Performances	Available resources	Occupied resources	% of utilization
Number of slice LUTS	5270	1328	1%
Number of slice register	11440	941	1%
BRAM	-	1	-

Table 3. Hardware analysis of proposed system for Virtex 5 device

Performances	Available resources	Occupied resources	% of utilization
Number of slice LUTS	28800	1475	1%
Number of slice register	28800	739	1%
BRAM	-	1	-

Table 4. Hardware analysis of proposed system for Spartan 6 device

Performances	Available resources	Occupied resources	% of utilization
Number of slice LUTS	9112	1139	1%
Number of slice register	18224	738	1%
BRAM	-	2	-

The hardware analysis of the proposed system for Virtex 4, Virtex 5 and Spartan 6 are shown in the Table 2, 3 and 4 respectively. Moreover, the delay performances of the FPGA devices are shown in the Table 5. The number of slice LUT, slice registers and BRAM of Virtex 4 are 1328, 941 and 1 respectively. Additionally, the number of slice LUT, slice registers and BRAM of Spartan 6 are 1139, 738 and 2 respectively. The proposed system delay for Spartan 6 device is 1ns, which is high when compared to the implementations of Virtex 4 and Virtex 5.

Table 5. Delay analysis for FPGA devices

FPGA devices	Virtex 4	Virtex 5	Spartan 6
Delay (ns)	0	0	1

Table 6. Failure rate analysis in terms of SNR

SNR	0dB	10dB	20dB
Proposed system	0.10%	0.07%	0.06%

Table 7. Failure rate analysis in terms of BER

BER	0	10%	20%
Proposed system	0.13%	0.17%	0.20%

Table 6 and Table 7 shows the failure rate analysis in terms of SNR and BER respectively. This Table 6 and 7 shows the amount of failure occurred in the Viterbi decoder. For example, the failure rate for 20 dB is 0.06% as well as the failure rate for 10% BER is 0.17%. This shows the developed system avoids the error approximately 99.9% while transmitting the data bits from the source to the destination.

Power analysis of proposed system: The static and dynamic power of the convolutional encoder and viterbi decoder of the proposed system is analyzed in this section. This power analysis is taken for different Intel FPGA families such as Cyclone II, Cyclone III, Cyclone IV, Stratix II and Stratix III.

The static and dynamic power analysis of the convolutional encoder and Viterbi decoder is shown in the Table 8. Moreover, the graphical illustration of the convolutional encoder and Viterbi decoder are shown in the Figure 6 and Figure 7 respectively. The Intel FPGA family is taken because of its value change dump file which used to provide input for dynamic power analysis. From the Table 8, knows that the static power is constant whereas the dynamic power is changed for an each device based on the triggered input. For example, the static and dynamic power of the Viterbi decoder in Cyclone II FPGA device are 58mW and 4.59mW respectively.

Comparative analysis: The performance of the proposed system is compared with DWC (Gao 2019) for knowing the effectiveness of the proposed method. In the ABFT is developed based on the improved structure of DWC. The improved DWC is used for providing the protection for the Viterbi decoder from SEUs. The comparative analysis is done in terms of number of slice LUT, number of slice register, BRAM, delay and failure rate. This DWC (Gao 2019) is implemented for different Intel FPGA families to evaluate the proposed system performances.

Table 8. Static and dynamic power analysis

Intel FPGA families	Convolutional encoder		Viterbi decoder	
	Static power (mW)	Dynamic power (mW)	Static power (mW)	Dynamic power (mW)
Cyclone II	58	2.67	58	4.59
Cyclone III	45	1.64	45	4.18
Cyclone IV	63	2.98	63	4.74
Stratix II	84	4.85	84	7.59
Stratix III	87	4.92	87	7.84

Figure 6: Static and dynamic power analysis for convolutional encoder

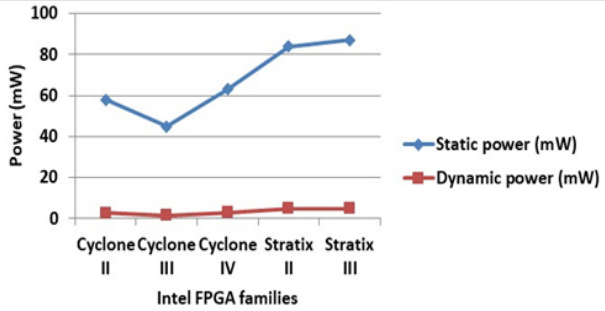


Figure 7: Static and dynamic power analysis for Viterbi decoder

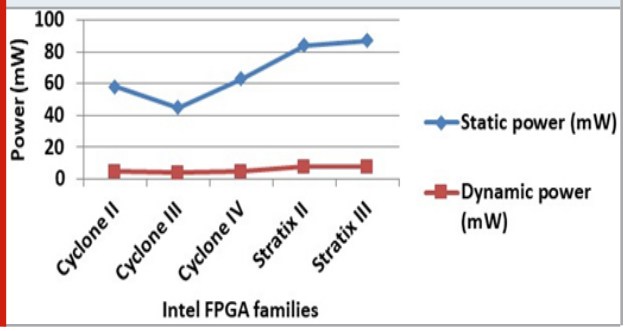


Table 9. Comparison of proposed system for hardware utilization and delay

Parameters	DWC (Gao 2019)	Proposed system		
		Virtex 4	Virtex 5	Spartan 6
Number of Slice LUTs	8672	1328	1475	1139
Number of Slice registers	5460	941	739	738
BRAM	2	1	1	2
Delay (ns)	1	0	0	1

Table 10: Comparison of Failure rate in terms of SNR

SNR (dB)	DWC(Gao 2019)	Proposed system
0	0.28%	0.10%
10	0.21%	0.07%
20	0.20%	0.06%

Table 11. Comparison of Failure rate in terms of BER

BER (%)	DWC(Gao 2019)	Proposed system
0	0.18%	0.13%
10	0.30%	0.17%
20	0.31%	0.20%

Table 9 shows the comparison of the hardware utilization and delay for the proposed system with DWC (Gao 2019). The failure rate comparison of proposed system with DWC (Gao 2019) is shown in the Table 10 and Table 11. The results of Table 10 and Table 11 are taken for the different SNR and BER levels respectively. Moreover, the power comparison for convolutional encoder and viterbi decoder with different Intel FPGA families are shown in the Table 12 and Table 13 respectively. From the Table 9, knows that the hardware utilization of proposed system is less when compared to the DWC (Gao 2019). For example, the proposed system slice LUT for Virtex 4 is 1328, it is less when compared to the DWC (Gao 2019). Moreover, the failure rate of the proposed system is less than the DWC (Gao 2019).

For example, the failure rate of the proposed system is 0.13%, it is less when compared to the DWC (Gao 2019). The static and dynamic power of the proposed system is less than the DWC (Gao 2019). The proposed system achieves better performance due to the FSAM based trellis integration in convolutional encoder and folding transformation in viterbi decoder. The FSM based trellis minimizes the complexity of the system that

helps to minimize the errors in the communication. The viterbi decoder with folding transformation reduces the number of stages processed during decoding process. The reduction in number of stages minimizes the area utilization throughout the system. Moreover, the area of the proposed system is directly proportional to the power consumption of encoder and decoder. Therefore, the static and dynamic power of the proposed system is less when compared to the DWC (Gao 2019).

Table 12. Comparison of convolutional encoder power for different Intel FPGA families

Intel FPGA families	Static power (mW)		Dynamic power (mW)	
	DWC (GAO 2019)	Proposed system	DWC (Gao 2019)	Proposed system
Cyclone II	62	58	3.12	2.67
Cyclone III	51	45	4.78	1.64
Cyclone IV	74	63	3.09	2.98
Stratix II	97	84	5.76	4.85
Stratix III	81	87	5.12	4.92

Table 13. Comparison of Viterbi decoder power for different Intel FPGA families

Intel FPGA families	Static power (mW)		Dynamic power (mW)	
	DWC (GAO 2019)	Proposed system	DWC (Gao 2019)	Proposed system
Cyclone II	62	58	5.84	4.59
Cyclone III	51	45	5.32	4.18
Cyclone IV	74	63	6.04	4.74
Stratix II	97	84	8.78	7.59
Stratix III	81	87	9.12	7.84

CONCLUSION

In this research, the Viterbi decoder is modified for reconstructing the encoded bits from the convolutional encoder. The convolutional encoder is combined with the FSM based trellis encoding to reduce the complexity in the proposed system. The Viterbi decoder is designed with the folding transformation for minimizing the amount of stages in the decoding process. This helps to reduce the hardware utilization in the wireless communication system as well as this proposed system effectively reconstructs the encoded data bits into the original input data bits. The proposed Viterbi decoder design is used to overcome the issues caused by the higher constraint length. Moreover, the proposed system gives better performance than the DWC in terms of hardware utilization, delay and failure rate. The proposed system's slice LUT is 1139 for Spartan 6, it is less when compared to the DWC.

REFERENCES

Ates, O., Keskin, S. and Kocak, T. (2016) High throughput graphics processing unit based Fano decoder. Journal of Network and Computer Applications, 75:128-137.

Bahrami, M. and Vasic, B. (2019) A deliberate bit flipping coding scheme for data-dependent two-dimensional channels. IEEE Transactions on Communications.

Broich, M. and Noll, T.G. (2017) Optimal Datapath Widths within Turbo and Viterbi Decoders for High Area-and Energy-Efficiency. Journal of Signal Processing Systems, 87(3):299-325.

Cai, X., Zhou, M. and Huang, X. (2017. Model-based design for software defined radio on an FPGA. IEEE Access, 5, pp.8276-8283.

El Chall, R., Nouvel, F., Hélard, M. and Liu, M. (2015) Iterative receivers combining MIMO detection with turbo decoding: performance-complexity trade-offs. EURASIP Journal on Wireless Communications and Networking, 2015(1):69.

Gao, Z., Zhu, J., Han, R., Xu, Z., Ullah, A. and Reviriego, P. (2019) Design and Implementation of Configuration Memory SEU-Tolerant Viterbi Decoders in SRAM-Based FPGAs. IEEE Transactions on Nanotechnology, 18:691-699.

Garcia-Herrero, F., Valls, J. and Meher, P.K. (2011) High-speed RS (255, 239) decoder based on LCC decoding. Circuits, Systems, and Signal Processing, 30(6), pp.1643-1669.

Kermani, M.M., Singh, V. and Azarderakhsh, R. (2016) Reliable low-latency Viterbi algorithm architectures benchmarked on ASIC and FPGA. IEEE Transactions on Circuits and Systems I: Regular Papers, 64(1):208-216.

Kim, J. and Sung, W., 2012. Low-energy error correction

of NAND Flash memory through soft-decision decoding. EURASIP Journal on advances in Signal Processing, 2012(1):195.

Kuang, S.R., Liang, C.Y. and Tseng, I.P. (2018) A Low-Power Codeword-Based Viterbi Decoder with Fine-Grained Error Detection and Correction Techniques. Arabian Journal for Science and Engineering, 43(2):585-595.

Leroux, C., Raymond, A.J., Sarkis, G., Tal, I., Vardy, A. and Gross, W.J. (2012) Hardware implementation of successive-cancellation decoders for polar codes. Journal of Signal Processing Systems, 69(3):305-315.

Liu, H., Jego, C., Boutillon, E., Jezequel, M. and Diguet, J.P. (2012) A contribution to the reduction of the dynamic power dissipation in the turbo decoder. annals of telecommunications-annales des télécommunications, 67(7-8):397-406.

Mathana, J.M., Rangarajan, P. and Perinbam, J.R.P. (2013) Low complexity reconfigurable turbo decoder for wireless communication systems. Arabian Journal for Science and Engineering, 38(10):2649-2662.

Marriwala, N., Sahu, O.P. and Vohra, A. (2013) 8-QAM software defined radio based approach for channel encoding and decoding using forward error correction. Wireless personal communications, 72(4):2957-2969.

Prakash, G. and Muthamizhse, I. (2016) FPGA Implementation of Bose Chaudhuri Hocquenghem Code (Bch) Encoder and Decoder for Multiple Error Correction Control. International Journal of Innovative Research in Science, Engineering and Technology, 5(3):4467-4473.

Reddy, B.S. and Rao, V.S.R. (2017) FPGA Implementation of LDPC Encoder and Decoder using Bit Flipping Algorithm. International Journal of Science and Research, 6(9):1683-1690.

Subramani, P., Rajendran, G.B., Sengupta, J., Pérez de Prado, R. and Divakarachari, P.B., 2020. A Block Bi-Diagonalization-Based Pre-Coding for Indoor Multiple-Input-Multiple-Output-Visible Light Communication System. Energies, 13(13), p.3466.

sSun, Y. and Cavallaro, J.R. (2011) A flexible LDPC/turbo decoder architecture. Journal of Signal Processing Systems, 64(1):1-16.

Vaithiyanathan, D., Nargis, J. and Seshasayanan, R. (2015) High performance ACS for Viterbi decoder using pipeline T-Algorithm. Alexandria Engineering Journal, 54(3):447-455.

Xie, Y.Q., Yu, Z.G., Feng, Y., Zhao, L.N. and Gu, X.F. (2018) A multistandard and resource-efficient Viterbi decoder for a multimode communication system. Frontiers of Information Technology & Electronic Engineering, 19(4), pp.536-543.

Zhang, W., Jia, Y., Meng, X., Brejza, M.F., Maunder, R.G. and Hanzo, L. (2014) Adaptive iterative decoding for expediting the convergence of unary error correction codes. IEEE Transactions on Vehicular Technology, 64(2), pp.621-635.