International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 18 (2017) pp. 7982-7989

High Speed Low Power Adaptive Viterbi Decoder Architecture for Underwater Acoustic Communication with Turbo Codes

Girish N[1]
Research Scholar, Department of Electronics and Communication, Visvesvaraya Technological University, BMS College of Engineering, Bengaluru-560019, Karnataka, India.

[1]Orcid Id: 0000-0002-5069-5854

Dr. Veena M B[2]
Associate Professor, Department of Electronics and Communication, BMS College of Engineering, Bengaluru-560019, Karnataka, India.

[2]Orcid Id: 0000-0001-9171-2625

Abstract

Underwater acoustic communication module based on Orthogonal Frequency Division Multiplexing (OFDM) uses rate ½ convolutional encoders and Turbo decoders for error control coding. Turbo decoders are designed with two Viterbi decoders that operate in sequence to improve Log Likelihood Ratio (LLR). The computation complexity of Viterbi decoders limits their use for high speed decoding as the decoders are iterative. In this paper, adaptive Viterbi decoder is designed that is based on novel message decoding logic, branch metric unit and survivor path metric unit. The computation complexity is reduced by replacing the arithmetic and logic unit with Look Up Table (LUT) thus increasing the access time. The adaptive decoder architecture eliminates the iteration process by predicting the state as well as the data and hence is faster in decoding message. The design is modeled and implemented on Virtex-5 FPGA and is estimated to operate at a frequency of 178 MHz consuming 1W of power and less than 10% of the LUT space.

Keywords: Underwater communication, Turbo Codes, OFDM, Viterbi Decoder, FPGA, High Speed.

INTRODUCTION

Increasing throughput in ocean propagation in highly reverberant channel is one of the major challenges being addressed by researchers in underwater acoustic communication & networking. The strong absorption of acoustic waves by water, multi-path fading, rapid time shifts and slow signal propagation causing Doppler effects limits communication bandwidth [1]. Orthogonal Frequency Division Multiplexing (OFDM) technology is advantages in terms of complexity and high data rate for Underwater Acoustic Communication (UAC) as compared with code division multiple access techniques [2]-[4]. Shallow underwater channels are modeled as Rayleigh stochastic processes based on ray theory multi-path model [5]. The Center for Maritime Research and Experimentation (CMRE) is in the process of promoting JANUS standards for modulation and coding in digital underwater communications. The recommendations JANUS standards are use of stronger error coding technique such as convolutional codes to overcome channel errors [6]. Forward error correction codes such as Turbo and Low-Density Parity Check (LDPC) are used for channel encoding. The bit errors in underwater channel are random due to ICI, ambient noise and impulsive noise. Turbo coding is preferred due to its flexibility in setting code rate by puncturing operation and reduced complexity as there is no need for matrix inversion operation as in LDPC [7]. Turbo decoding is carried out using soft-demapping of information by computing Log Likelihood Ratio (LLR). The coding gain in Turbo decoder is achieved by use of Viterbi Decoder (VD) algorithm [8]. In [9], Turbo encoder and decoder is used for underwater acoustic communication channel with fractional cosine transform. In [10] Turbo encoder and decoder is used for underwater acoustic channel with 8 PSK and 16 QAM. The Turbo decoder comprises of two channel decoders based on Viterbi decoding algorithm to compute LLR. Most of the literature report on modelling and analysis of underwater communications in software environment. One of the important recommendations of JANUS standards is to deploy the algorithms on existing platforms such as DSP or FPGA hardware.

Viterbi decoder which is an integral part of Turbo decoder needs to be implemented with reduced computation complexity. In order to reduce computation complexity and reduce decoding time adaptive Viterbi algorithm was developed by Chan and Haccoun [11], to reduce the number of computations in decoding message from received information with comparable Bit Error Rates (BER). In [12] dynamically reconfigurable adaptive Viterbi decoder was proposed and implemented on Xilinx FPGA, with dynamic reconfiguration rate of 130 Kbps and 100% LUT utilization. In [13] low-power Viterbi decoder design based on scarce state transition (SST) is presented that operates at 100MHz

clock with data rate of 160 Mb/s, consuming power less than 198 mW, occupying chip area of 3.7 mm^2 using 0.18 micron CMOS technology. In [14] the VD with the 2-step pre-computation architecture and one with the conventional *T*-algorithm are modeled that operates at 447 MHz, with chip area of 0.58 mm^2 and power consumption of 20mW. In [15] Viterbi decoder with constraint length of 7, code rate of 1/2 is implemented on Virtex-II FPGA that operates at 80Mbps, occupying 10% of chip area and reducing power dissipation by 5%. The VDA uses trace back method for decoding and hence requires more time for decoding, the decoder is tested for error bits introduced every 10^{th} bit. In this work, modified adaptive Viterbi decoder architecture is proposed that is based on parallel decoding architecture with direct decoding algorithm with novel architecture for branch metric computation, path metric computation and add-compare-select unit.

VITERBI DECODER ALGORITHM

Input data encoded using convolutional encoder creates code symbols of *n* from *k* input symbols, thus defining the code rate as *k/n*. Convolutional encoder with rate k/n has a constraint length K. Transitions between states with time is described as Trellis diagram which is used to compute the output and the next states for a given input. In the trellis diagram S denotes the states and C denotes the encoded outputs, as the state transition occur from present sate t-1 to next state t. The index for stat transitions are valid for every j, $0 \leq j \leq 2^{K-1}-1$. Constraint length K (that indicates the number of times the input bit influences in producing output bits) decides the number of states N in trellis diagram ($N=2^{K-1}$). The encoded data with noise is decoded with Viterbi algorithm which is based on Maximum Likelihood path through the trellis that has the largest log-likelihood function by Andre et al [16] as in Eq.(1), (Cm is encoder output for a given path m and Y is the received signal).

$$\ln[(P(Y|Cm)] = \sum_{n'}^{\infty} \ln[P(Yn'|Cmn')] \quad (1)$$

The block diagram of Viterbi decoder is as shown in Figure 1, it consists of Branch Metric Unit (BMU), Add-Compare-Select Unit (ACSU) and Survivor Memory Unit (SMU). The BMU computes Euclidean distance between the received sequence (Y1 and Y2) and probable codewords associated with a branch (C1 and C2), for rate R=1/2, the BM can be represented as in Eq. (2)

$$BM = (Y1 - C1)^2 + (Y2 - C2)^2 \quad (2)$$

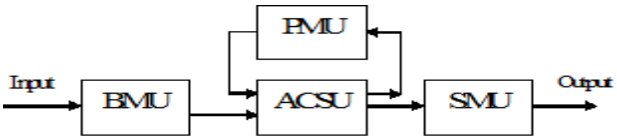

Figure 1: Block diagram of Viterbi decoder

Accumulation of BM along a path is termed as Path Metric (PM), state metric is the optimal paths of state S from the initial time to current time t_N. At every state there would be two competing paths, the decoder selects the path with minimum PM. The ACSU operation that performs the selection with path metric update is as in Eq. (3),

$$PM_{Sj}^{(n)} = min\left\{PM_{S2j}^{(n-1)} + C_{2j},\ PM_{S2j+1}^{(n-1)} + C_{2j+1,j}\right\}$$
$$PM_{Sj+2}^{(n)} = min\left\{PM_{S2j}^{(n-1)} + C_{2j,j+2},\ PM_{S2j+1}^{(n-1)} + C_{2j+1,j+2}\right\} \quad (3)$$

The selection of minimum path [7] is accomplished by the decisions $D_{Sj}^{(n)}$ and $D_{Sj+2}^{(n)}$ as in Eq. (4)

$$D_{Sj}^{(n)} = sign\left\{PM_{S2j}^{(n-1)} + C_{2j} - PM_{S2j+1}^{(n-1)} + C_{2j+1,j}\right\}$$
$$D_{Sj+2}^{(n)} = sign\left\{PM_{S2j}^{(n-1)} + C_{2j,j+2} - PM_{S2j+1}^{(n-1)} + C_{2j+1,j+2}\right\} \quad (4)$$

The decoding algorithm performs computation of PM for a series of time steps termed as *truncation length* (T_L), which is also known as the minimum distance path that is computed by identifying the most-likely transmitted symbol sequence. In [17] the truncation length is given by Eq. (5), where R is the coding rate, C is the channel capacity and K is the constraint length, the maximum truncation length is 64.

$$\frac{T_L}{K} \geq \begin{cases} \frac{1}{1-\frac{2R}{C}}, & 0 \leq R \leq \frac{C}{4} \\ \frac{1}{2\left(1-\sqrt{\frac{R}{C}}\right)^2}, & \frac{C}{4} \leq R \leq \frac{C}{2} \\ \frac{\left(1+\sqrt{\frac{R}{C}}\right)}{\left(1-\sqrt{\frac{R}{C}}\right)}, & C/2 \leq R < C \end{cases} \quad (5)$$

One of the major limitations in Viterbi decoder is the implementation complexities as the decoder need to retain 2K-1 trellis states for every symbol received. For larger constraints length the memory requirement grows exponentially as the number of paths stored at every stage increases thus adding to computation complexity. Instead of retaining all 2K-1 paths, the computation complexity and memory requirement can be reduced by retaining only the paths that satisfy important criteria that are predetermined. In AVD, there are three parameters that decide the number of paths to be retained is based on T, d_m and N_{max}. T is the

threshold (identified by the designer), d_m is the Hamming distance of the survivor paths at level (l-1) and N_{max} is the total number of survivor paths that is selected by the designer. As the decoding algorithm moves from level (l-1) to level l, the hamming distance that are equal to or less than (T + d_m) are retained. Further the number of survivor paths is retained if they do not exceed N_{max}. A path at every stage of decoding is retained by using the hamming distance at the previous level instead of the minimum hamming distance at the current level. Figure 2 shows the trellis diagram for ADA for states from time t_0 to t_{11}.

In AVD, the path metric error and the number of paths N_{max} are used to reduce the number of surviving paths that can progress to next state in the trellis. For example at time t_4, dm is computed to be 0, and hence dm+T is 1, the number of states reaching t_4 are 4. As the threshold is at 1, the paths 1, 2 and 3 are discarded with path metric error (2), (3) and (2) respectively retaining the 4th path with path metric (1) reaching the state S_3. At every stage of decoding the surviving path with minimum length is retained and stored in memory.

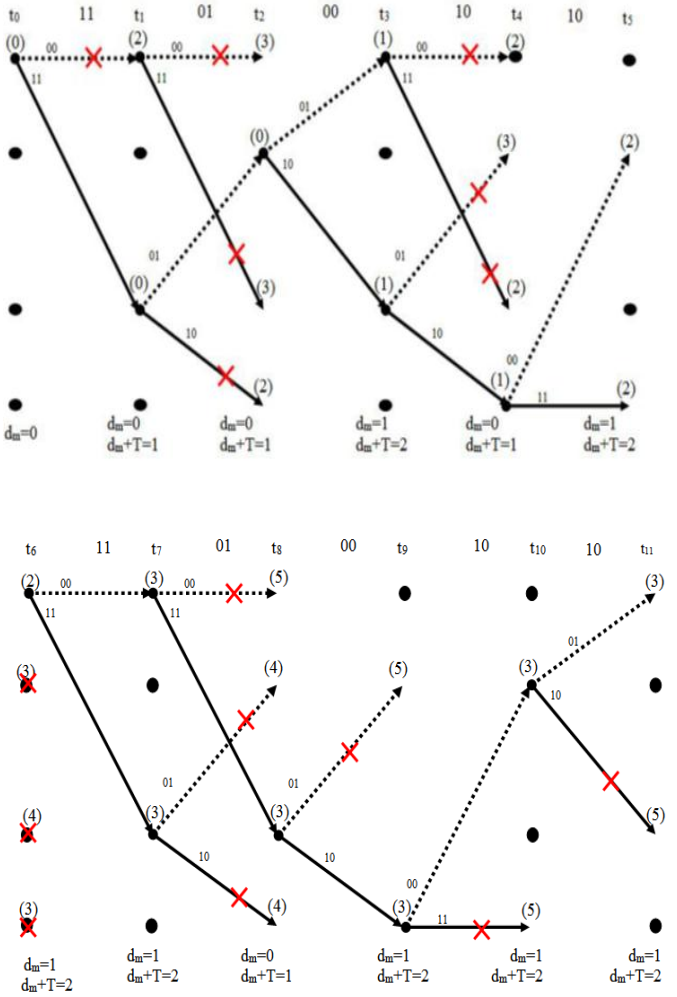

Figure 2: Adaptive Viterbi algorithm (trellis states $t_0 - t_{11}$)

Dynamically Adaptive Viterbi decoder architecture

In the proposed architecture for dynamically adaptive Viterbi decoder architecture, the threshold level dynamically changes its value according to Eq. (6). If the truncation length is t_N, consisting of N-1 trellis states, requiring N-1 time slots for decoding, the threshold values are set as in Eq. (6),

$$threshold\ T = \begin{cases} dm + T, & 0 \le t < t + i \\ dm + 2T, t + i \le t < t + 2i \\ dm + 3T, t + 2i \le t < tN \end{cases} \quad (6)$$

Where *i* is integer (*i* = 7 is selected in this work), the selection of *i* depends on number of input bits being encoded, convolutional encoder and truncation length. The threshold T is 2 for time units 1 to 6, and from time unit 7 onwards the threshold is set to 3. At every trellis state, the Path Metric (PM) and the Survivor State (SS) are computed from the ACS unit, the path metric from every state is fed back into the PMU FIFO register for next iteration. From the survivor states and the path metric the path with minimum error is retained. At time unit 6, in the PM is 2, 3(-), 4(-), 3(-) for S0, S1, S2 and S3 states respectively. The PM '-' indicates that the path does not exist or is terminated as per the adaptive Viterbi algorithm.

Figure 3 shows the adaptive Viterbi decoder architecture for trellis state S0. At time tn, C1 and C2 are the expected outputs at state S0 (00, 01 respectively), these are XOR with received code word (R) to compute the branch metric. The branch metric is added with the path metric corresponding to the two paths reaching S0.The compare unit selects the paths with lowest path metric.The selected path metric is stored in the path metric FIFO, also the selected path (either S0 or S1) is selected based on the comparator output and corresponding input which is 0 is stored in the decoded message FIFO. The path metric is fed back into the ACS unit for computation of next state path metrics and the decoded message for new set of received code word.

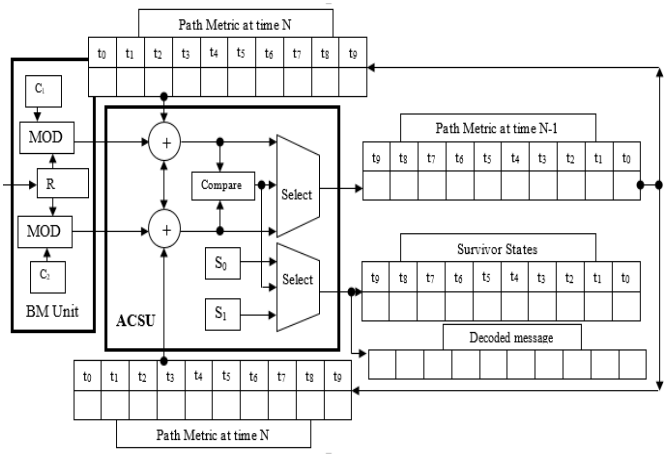

Figure 3: Adaptive Viterbi Decoder Architecture

Table 1: Decoded message from PMs and proposed logic

Time(t)/ States(S)	1	2	3	4	5	6	7	8	9	10	11
S0			0			0	0				0
S1		0			0					0	
S2	1		1					1			
S3				1	1		1		1		-

The adaptive algorithm presented in this paper is straight forward in computing the outputs from the PMs computed at every state, hence does not required additional memory and trace back logic for decoding message. Table 1 shows the decoded message from PMs computed based on the logic proposed.

Subsystems design for AVD architecture

BMU and ACSU operations that form the critical subsystems are designed using only memory based approach. At every state the expected output code is known and is presented in Table 1, with new inputs received from the channel, it is required to compute branch metric. In [18], BM unit consists of adders and registers, in these work adders are eliminated based on memory mapping logic. Received code word which is of two bits is of four possible combinations and the BM computation at every node is computed based on the logic discussed in Table 2.

Table 2: Possible branch metric output at node 0 (State S0)

Received code word	Expected output at node 0 (S0)	Branch metric
00	00	00
01	00	01
10	00	01
11	00	10
Received code word	Expected output at node 0 (S1)	Hamming distance
00	01	01
01	01	00
10	01	10
11	01	01

Figure 4 shows the proposed branch metric unit architecture that consists of input registers that store the expected outputs C at each node that are concatenated to form four bit address with the arrival of received code word (r). The memory unit of that stores the precomputed BM is read into the ACS unit according to the memory address.

The adder unit and the compare unit in the ACSU are replaced with LUT logic. The adder unit receives the output from BM unit and the PM unit that are added to compute new PM. BM unit has three possible outputs the PM unit in adaptive Viterbi decoder has four possible PMs {0, 1, 2, 3}. The BM output (2 bits {0, 1, 2}) and the PM outputs (maximum 3 bits {0, 1, 2, 3}) are concatenated to form a five bit address to the memory. The precomputed adder results stored are read out into the comparator unit. The comparator unit is also realized using LUT approach. The outputs of two adder units can have the possible values of {0, 1, 2, 3, 4, 5} hence considering all possible comparisons there are 34 locations for the example considered, the outputs of PM unit each are of 3 bits are concatenated to from a six bit address to the comparator memory. The comparator memory is stored with 0 and 1, indicating whether the PM0 or PM1 is greater or less. If PMs are equal then priority is given to PM0 and the output stored in the corresponding location is 0. Figure 5 shows the select unit and the PM storage unit that selects the path metric with minimum distance and is stored in the FIFO. The path metric unit is fed back into the adder unit for next stage computation. The path metric unit from node 0 to node 3 at every stage of trellis is read into a message decoder unit.

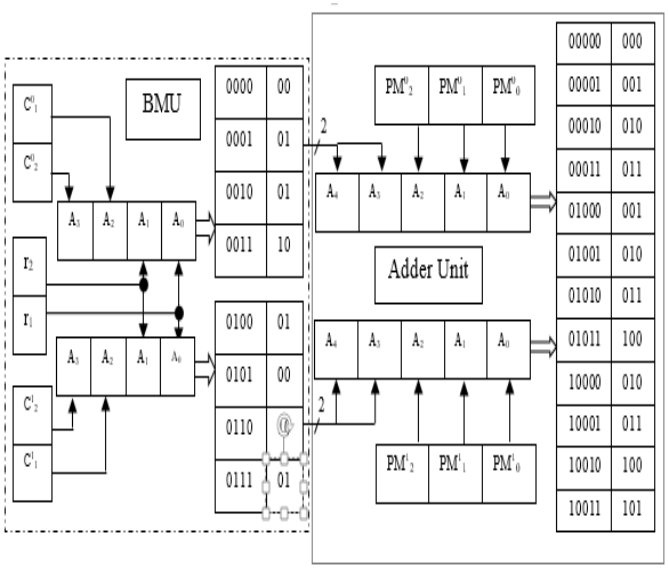

Figure 4: Proposed BMU and Adder Unit

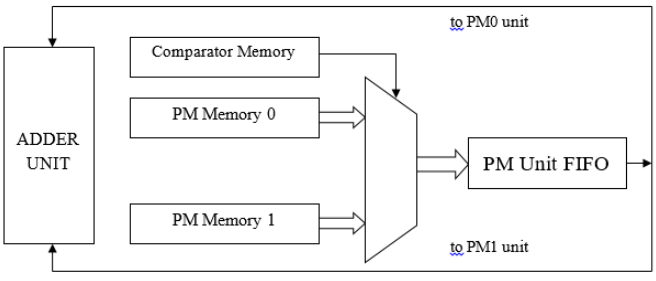

Figure 5: Select unit with PM FIFO

Figure 6 shows the path metric selection unit with Message Decoding Unit (MDU) for adaptive Viterbi decoder. The comparator selects the minimum path metric and corresponding message is decoded and stored in the message decoder FIFO.

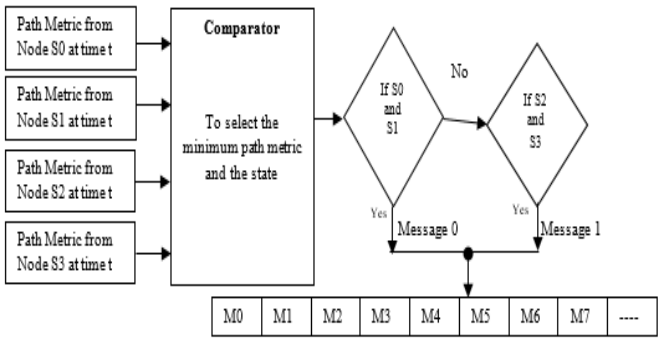

Figure 6: Message decoder algorithm for adaptive Viterbi decoder

Novel message decoding architecture for AVD

In order to improvise the performance in terms of hardware resources, novel architecture for message decoding is discussed. The adaptive architecture presented in the previous section is modified to improve the decoding speed. In [19] a novel architecture for Add-Compare-Select (ACS) unit is proposed for adaptive Viterbi decoder implemented using 180nm CMOS technology that reduces storage space, power dissipation and delay. The power dissipation and delay can be further reduced with design of Branch Metric Unit (BMU) and Survivor Memory Unit (SMU). In [15] Viterbi decoder with constraint length of 7, code rate of 1/2 is implemented on Virtex-II FPGA that operates at 80Mbps, occupying 10% of chip area and reducing power dissipation by 5%. The VDA uses trace back method for decoding and hence requires more time for decoding, the decoder is tested for error bits introduced every 10^{th} bit. In [20] Register Exchange (RE) method is adopted for Viterbi decoder with reduced memory storage with power reduction of 23%, BER of 10^{-5} and SNR of 6.3dB. The decoder complexity increases with increase in constraint length, use of adaptive Viterbi decoder could reduce computation complexity. A modified trace back method for Viterbi decoder is implemented on Virtex-II FPGA operating at 6 Mbps and occupying 500 slices. The part parallel architecture is suitable for medium speed and delay applications such as satellite communications. Systolic array architecture with pipelining is used to realize Viterbi decoder that reduces dynamic power by 43% with increased hardware resources [21]. The decoder algorithm uses trace back method for decoding and hence occupies more decoding time, which can be reduced using RE method. Dynamically reconfigurable systolic array based trace back method Viterbi decoder is implemented in [18] on NEC electronics processor. The reconfigurability is achieved for five different decoders dynamically, for constraint length 3 to 7 is suitable for mobile communication, for Underwater communication more generic architecture is required.

In this paper, a novel architecture for BMU, ACSU and Message decoder unit is proposed and implemented on FPGA utilizing the hardware resources effectively and is suitable for high data rate applications. Message decoding is either based on register exchange method and trace back method based on the data in the survivor path memory.For the first time direct method for message decoding based on the survivor paths and path metrics is proposed. The path metrics of four nodes the path metric with minimum error is retained, in order to compare the path metric of four nodes a comparator is used as shown in Figure 6. A novel architecture for comparator is designed in this paper. The path metrics for the first stage (first six clock units) as discussed does not exceed a value more than 4, similarly for the second stage (from clock unit seven to clock unit 13) path metric would not exceed more than 6, hence the path metric at every node will be represented by a three bit number PM = $\{PM^0_0, PM^0_1, PM^0_2\}$. The novel architecture consists of a equality operator that compares the PM of four nodes with known possible path metrics of {1, 2, 3, 4, 5, 6} as shown in Figure 7 consisting of six parallel processing units with each unit consisting of four equality check unit and one four input OR gate. The output of OR gate is for the first three configurations are represented as in Eq. (7),

$$E_1 = \text{OR}\ \{PM0 = 001, PM1 = 001, PM2 = 001, PM3 = 001\},$$
$$E_2 = \text{OR}\ \{PM0 = 010, PM1 = 010, PM2 = 010, PM3 = 010\},$$
$$E_3 = \text{OR}\ \{PM0 = 011, PM1 = 011, PM2 = 011, PM3 = 011\}, \quad (7)$$

Output of OR gate (E_1) is '1' if the PM at one of the four nodes is 1, E_2 is '1' if the PM at one of four nodes is 2, similarly if one of the four PMs is 6, E_6 is '1'. The output E1 to E6 are priority encoded with highest priority set to E_1, indicating that the lowest path metric out of four PMs is 1. The Minimum Path Metric Compute unit (MPMC) shown in Figure 7 generates Priority Encoder PE output (6-bit) along with four bit equality check output (from six nodes). To decode the message output the six bit PE output and four bit equality check output are processed in the Message Decoder Unit (MDU) as shown in Figure 8. The PE signal selects the one of the four bit outputs from equality check unit out of six such inputs. The four equality check signals are used as selected lines to choose between one of the nodes {S0, S1, S2, S3}. As discussed in the previous section the output is '0'if the nodes are S0 and S1, else '1' if nodes are S2 and S3. The decoded message is stored in the message storage unit

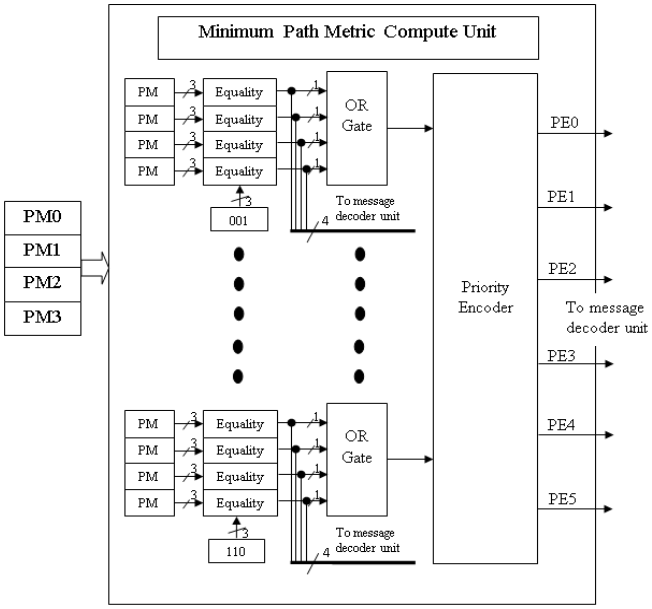

Figure 7: Minimum path metric compute unit

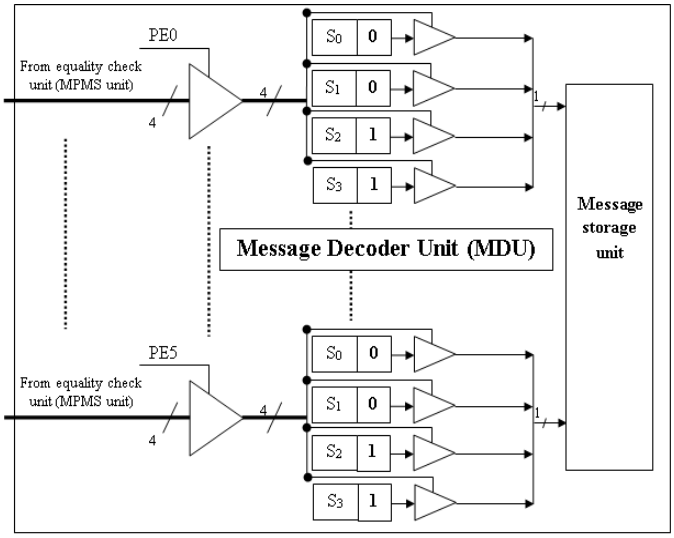

Figure 8: Message decoder unit

RESULTS AND DISCUSSION

The proposed AVD architecture with message decoding which forms the core is implemented on FPGA to evaluate its performance. The functionally correct HDL model for the AVD architecture is synthesized using Xilinx ISE targeting Virtex-5 FPGA. The RTL code is synthesized using Xilinx ISE targeting Virtex-5 FPGA with 110 million gate capacity. The architecture consists of 69120 slices and LUTs, 1014 number of Input Output Blocks (IOBs) and DSP block sets. As Viterbi decoder is one of the sub systems in underwater communication system, for System on Chip (SoC) design, Viterbi decoder should occupy minimum number of resources on FPGA providing space for all other modules. The design constraints are set to default settings and the synthesis is carried out. RTL code for AVD architecture (reported in literature) is developed and synthesized for comparison. Physical design is also carried out to perform placement and routing. The results obtained are based on default constraints set in Xilinx ISE tool, to improve the operating frequency further and optimize area resources advanced constraints are applied.

The optimization goal is set to speed compared instead of area, optimization effort is set to high instead of normal and power optimization is also given priority. In order to ensure that the hierarchy in Viterbi decoding algorithm is not flattened (this helps in dynamic reconfiguration). As there are large number of LUT-FF pairs (1014), synthesis constraint is set to utilize these pairs instead of DSP block sets. Virtex-5 architecture has CLBs that is made up of 6-input LUTs and hence it is appropriate to use LUTs rather than DSP block sets. The synthesis tool by default prefers DSP blocks for arithmetic operations and memory blocks for storage that are additional resources outside the CLBs. For proper utilization of CLB resources, it is required to specify the constraint, and in this work the LUT-FF utilization ratio is set to 100%, DSP utilization ratio is set to 30% and Block RAM utilization ratio is set to 50%. The case statements used in Verilog model are set be realized using fully parallel architecture instead of default scheme. Performances of the proposed architecture are compared with references and are tabulated in Table 3. From the comparisons, the proposed architecture has higher data rate and also consumes less memory and hence is suitable for underwater acoustic communications that requires high data rate. Power dissipation in the proposed design is due to large number of resources that are unused that consumes leakage power.

Table 3: Comparison of proposed AVD with reference work

	Reference [11]	Reference [12]	Reference [15]	Reference [21]	Proposed AVD
Device	XC2V2000 fg676	XC4036XL-08	EP1S20F484/ APEX EP20K200	XC2V1000-4FG256	XCV5110tff1136
Data rate	80 MHz, 80 Mbps	160.8 Kbps	102MHz/ 40MHz, 588 Kbps	40 MHz 78 Kbps	178.91MHz 180 Mbps
Resource utilization	316 out of 10752 (Slices)	1215 (CLBs)	14226/ 2200 (Logic elements)	2934 Slices	1011 out of 69120 (LUTs)
Power dissipation	59 mW	--	---	473 mW	1.027 W

CONCLUSION

In this paper modified adaptive Viterbi decoder architecture with message decoding logic and modified adaptive decoder architecture is presented. The proposed architecture is

designed and modeled using HDL and implemented using Virtex-5 FPGA. The proposed architecture is first of its kind for adaptive Viterbi decoding and has been design using optimum methods to improve decoding speed and minimum area. As the architecture is adaptive and has been implemented on FPGA platform they are reconfigurable. Dynamic reconfigurability can be achieved by designing the ACSU and SMU as compatible with MDU.

REFERENCES

[1] Mandar Chitre, Shiraz Shahabudeen, Lee Freitag, Milica Stojanovic., "Recent Advances in Underwater Acoustic Communications & Networking," Spring, Vol 42, No.1, pp.103-116, 2008.

[2] Li, H., Wang, X., Zou, Y., 2014 "Dynamic subcarrier coordinate interleaving for eavesdropping prevention in OFDM systems," *IEEE Commun. Lett.*, 18, (6), pp. 1059–1062, 2014.

[3] Li, H., Wang, X., Zou, Y., *et al.* "Eavesdropping-resilient OFDM system using CSI-based dynamic subcarrier allocation". Proc. of IEEE 77th Vehicular Technology Conf. (VTC Spring), pp. 1–5, 2013.

[4] *Chunguo Li1, Kang Song1,2, Luxi Yang1.,* "Low computational complexity design over sparse channel estimator in underwater acoustic OFDM communication system," *IET Commun.*, Vol. 11 Iss. 7, pp. 1143-1151, 2017.

[5] M. Chitre., "A high-frequency warm shallow water acoustic communications channel model and measurements, *The Journal of the Acoustical Society of America,* vol. 122, pp. 2580-2586, 2007.

[6] Joao Alves, Thomas Furfaro, Kevin LePage, Andrea Munaf`o, Konstantinos Pelekanakis, Roberto Petroccia, Giovanni Zappa, Moving JANUS forward: a look into the future of underwater communications interoperability.

[7] Tran Minh Hai, Underwater Wireless Communication using Acoustic OFDM, PhD Thesis, Graduate School of Engineering and Science,university of the ryukyus, August 2016

[8] T. K. Moon, Error Correction Coding: Mathematical Methods and Algorithms. Wiley, pp.181-192, 2005.

[9] Chen, Yixin and Clemente, Carmine and Soraghan, John and Weiss, Stephan (2015) Fractional Cosine Transform (FrCT)-Turbo based OFDM for underwater acoustic communication. In: Sensor Signal Processing for Defence (SSPD), 2015. IEEE, pp. 147-151. ISBN 9781479974443 http://dx.doi.org/10.1109/SSPD.2015.7288503

[10] Yuri Labrador, Masoumeh Karimi, Deng Pan, and Jerry Miller., "Modulation and Error Correction in the Underwater Acoustic Communication Channel," **IJCSNS** International Journal of Computer Science and Network Security, Vol.9, No.7, 2009.

[11] Chan, F, Haccoun, D, 'Adaptive Viterbi Decoding of Convolutional Codes over Memoryless Channels', IEEE Transactions on Communications, vol. 45, no. 11, pp. 1389–1400,1997.

[12] Sriram Swaminathan, Russell Tessier & Dennis Goeckel, 'A dynamically reconfigurable adaptive viterbi decoder', Proceedings of the International Symposium on Field Programmable Gate Arrays Archive, pp. 227-236,2002.

[13] Jie Jin, Chi-ying Tsui, 'Low-Power Limited-Search Parallel State Viterbi Decoder Implementation Based on Scarce State Transition', IEEE transactions on very large scale integration (VLSI) systems, vol. 15, no. 10, pp. 1172-1176,2007.

[14] Jinjin He, Huaping Liu, Zhongfeng Wang, Xinming Huang, Kai Zhang, 'High-Speed Low-Power Viterbi Decoder Design for TCM Decoders', IEEE transactions on very large scale integration (VLSI) systems, vol. 20, no. 4, pp.755-759,2012.

[15] Yan Sun, Zhizhong Ding, FPGA Design and Implementation of a Convolutional Encoder and a Viterbi Decoder Based on 802.11a for OFDM, Wireless Engineering and Technology, vol. 3, pp. 125-131, 2012.

[16] Andre Neubauer, Jurgen Freudenberger, Volker Kuhn, Coding theory: algorithm, architectures and applications, John Wiley & Sons Ltd, 2007.

[17] Lin, Y, Appenzeller, J, Knoch, J, Avouris, P, 'High-performance carbon nanotube field-effect transistor with tunable polarities', IEEE Trans. Nanotechnology, vol. 4, no. 5, pp. 481–489,2005.

[18] Shohei Abe, Yohei Hasegawa, Takao Toi, Takeshi Inuo, Hideharu Amano, 'An Adaptive Viterbi Decoder on the Dynamically Reconfigurable Processor', IEEE International Conference on Field Programmable Technology, pp. 285-288,2006.

[19] Yao Gang, Tughrul Arslan, 'A Novel VLSI Architecture for ACS Operation in Adaptive Viterbi Decoding', International Conference on Communications, Circuits and Systems (ICCCAS 2004), vol. 2, pp: 1434 – 1437, 2004.

[20] Dalia A. El-Dib, Mohamed I. Elmasry, 'Modified Register-Exchange Viterbi Decoder for Low-Power Wireless Communications', IEEE transactions on circuits and systems—I: regular papers, vol. 51, no. 2, pp. 371-378,2004.

[21] Man Guo, Omair Ahmad, M, Swamy, MNS, Chunyan Wang, 'FPGA Design and Implementation of a Low-Power Systolic Array-Based Adaptive Viterbi Decoder', IEEE Transactions on circuits and systems—I: regular papers, vol. 52, no. 2, pp. 350-365, 2005.