ISSN No:-2456-2165

Low Power Design of Adaptive Viterbi Decoder

Sujatha Cyri, Dharmistan K. Varugheese

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.

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

I. 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 Researchand 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 operationas 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 & 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 comp utation and add-compare-select unit.

II. 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

ISSN No:-2456-2165

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 asin Eq. (2)

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

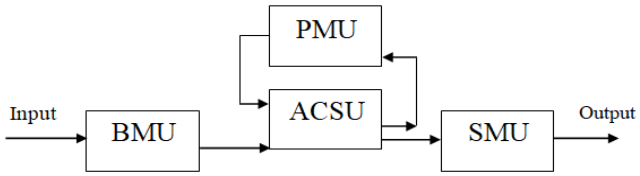

Fig. 1 Block diagram of Viterbi decoder

The adaptive algorithm presented in this work 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.

III. 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 know and is presented in Table 1, with new inputs received from the channel, it is required to compute bra nch metric. As discussed 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 1.

Table 1 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 2 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.

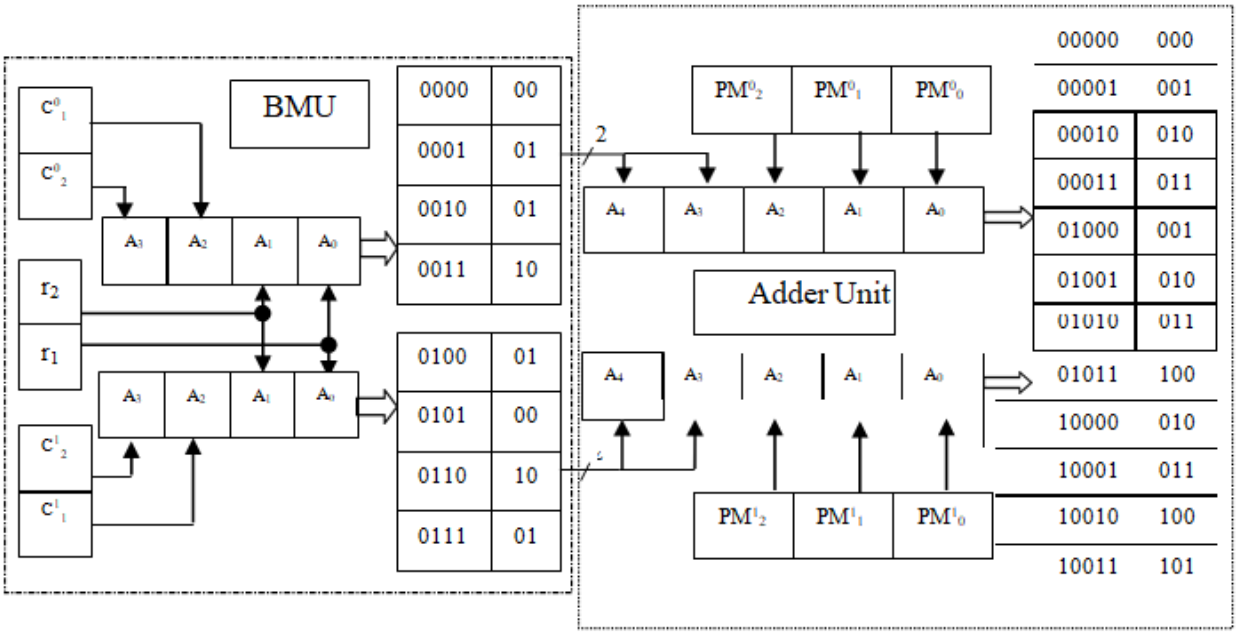

Fig. 2 Proposed BMU and Adder Unit

Figure 3 shows the path metric selection unit with Message Decoding Unit (MDU) foradaptive Viterbi decoder.

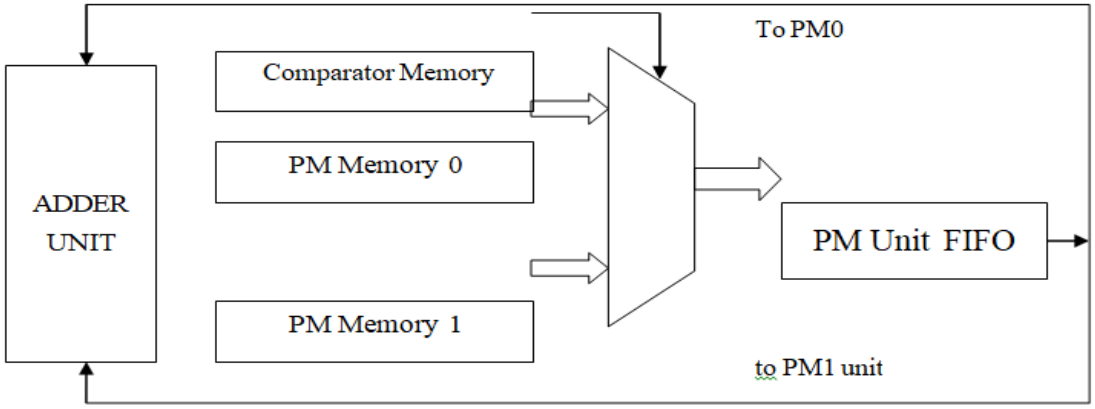

Fig. 3 Select unit with PM FIFO

The comparator selects the minimum path metric and corresponding message is decoded andstored in the message decoder FIFO as shown in Figure 4.

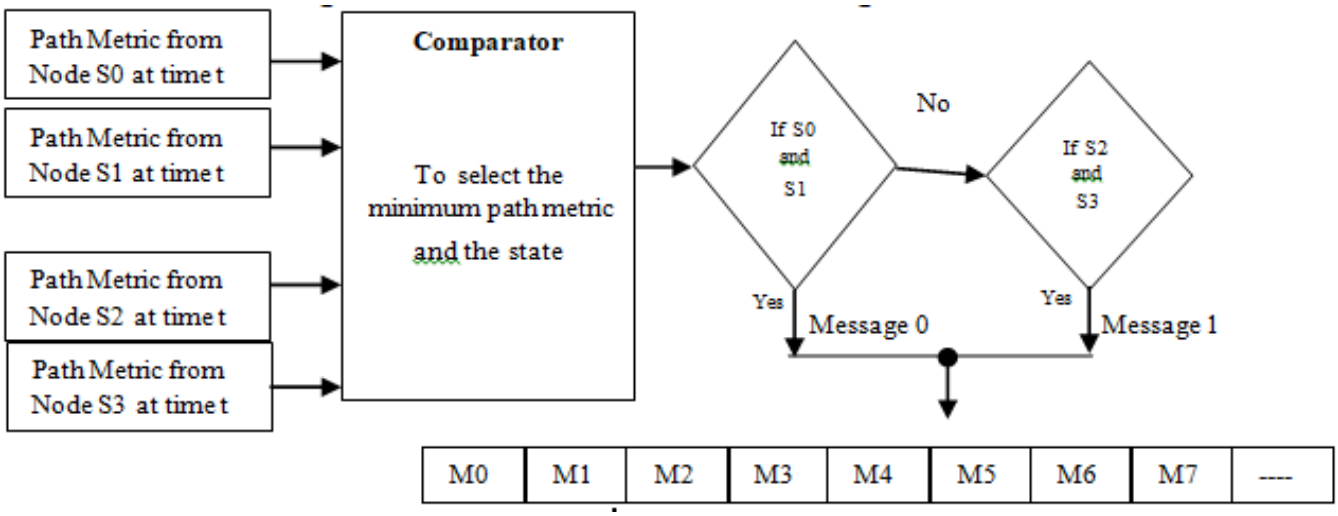

Fig. 4 Message decoder algorithm for adaptive Viterbi decoder

ISSN No:-2456-2165

IV. NOVEL MESSAGE DECODING ARCHITECTURE FOR AVD

In order to improvise the performance in terms of hardware resources, novel architect ure for message decoding is discussed in this section. 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 Wireless communication more generic architecture is required. In this work, a novel architecture for BMU, ACSU and Message decoder unit is proposed and implemented onFPGA 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. In this paper 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 fournodes a comparator is used as shown in Figure 4. A novel architecture for comparator is designed in this work. 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 }. 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 5 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 = OR {PM0 = 001, PM1 = 001, PM2 = 001, PM3 = 001},E_2 = OR {PM0 = 010, PM1 = 010, PM2 = 010, PM3 = 010},
E_3 = OR {PM0 = 011, PM1 = 011, PM2 = 011, PM3 = 011}, (7)

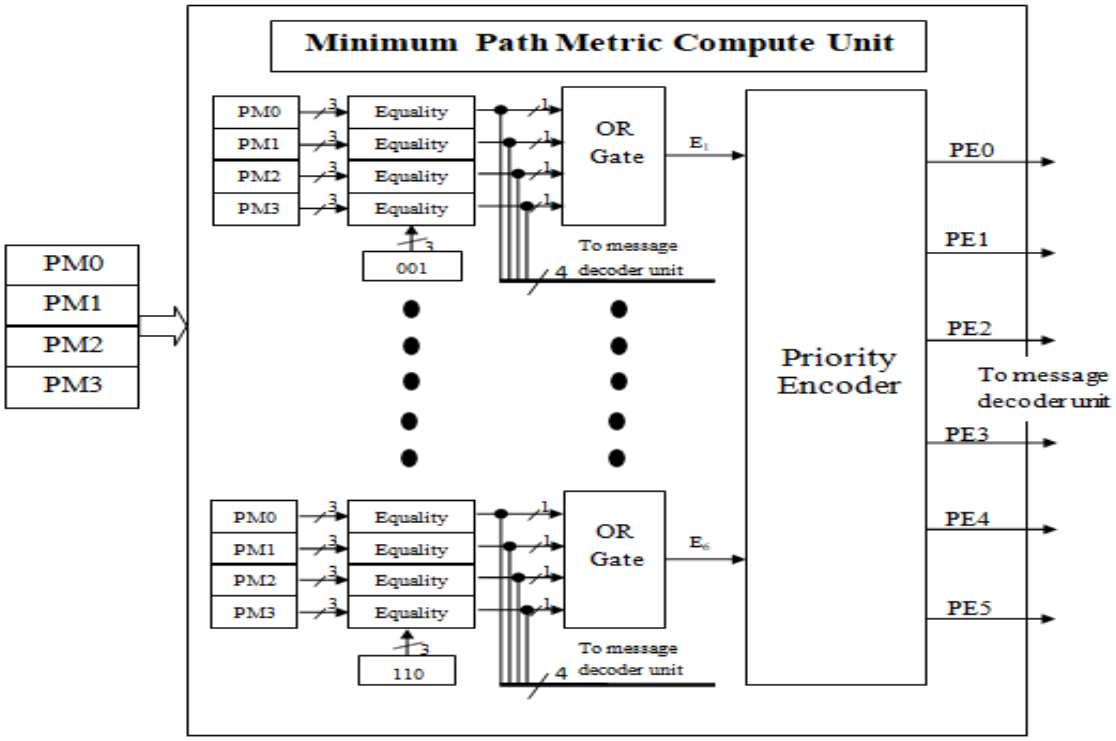

Fig. 5 Minimum path metric compute unit

ISSN No:-2456-2165

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 5 generates PEn output (6-bit) along with four bit equality check output (from six nodes).

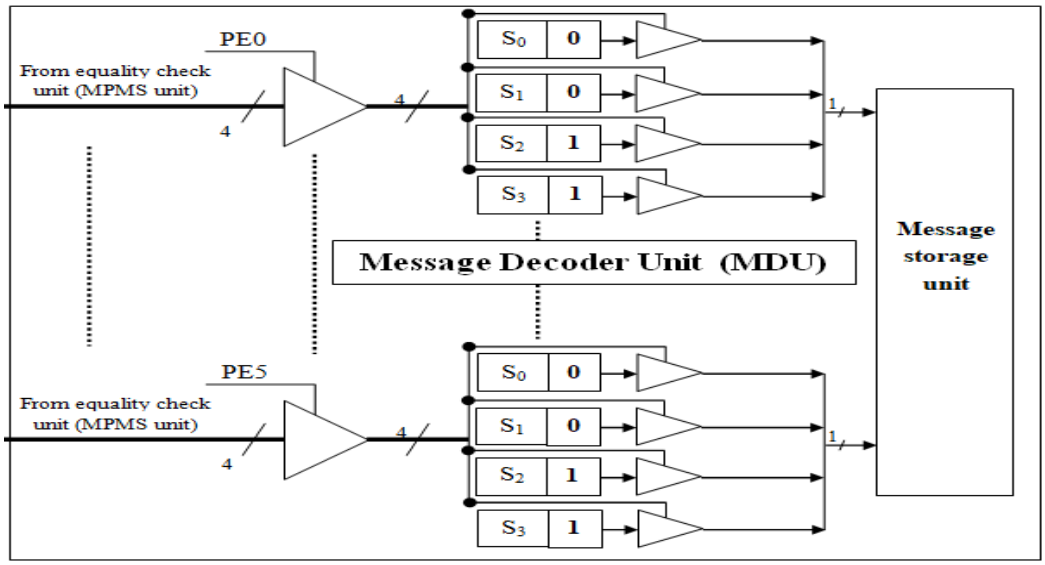

Fig. 6 Message decoder unit

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 6. The PEn 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. The proposed adaptive Viterbi decoder architecture with direct decoding logic is modeled using HDL and implemented using ASIC design flow, synthesis constraints are set to optimize area and maximum utilization resources. Low power design of AVD is presented in next section.

V. LOW POWER DESIGN

Synthesis and physical implementation was carried out for all below mentioned combinations. These were accomplished by changing the target library during compilation and in physical implementation by providing different reference libraries.

1. Multi height flow with different cell height
2. High Vt to Multi Vt flow
3. Mvt flow
4. Lvt flow
5. Hvt flow
6. Lvt to Hvt (or Mvt) flow
7. Rvt flow
8. High Vt to Multi Vt flow for netlist with DFT
9. Multi Vt flow for netlist with DFT

Same constraints were used for all the above combinations. The number of clock domain is one and clock period is 4ns. Input delay and output delay constraints are also there. There is no false path or multicycle path. Different cell height libraries do not affect synthesis process and optimization. But special care has to be taken for the p lacement of multi height cells. The unit tile height of lvt cells is 2. 52 µ and hvt cells are 1.96 µ. Hence two separate unit tiles have to be created and should be added in the technology file. Hvt reference library is created with the unit tile name "unit" and lvt reference library is created with unit tile name "lvt_unit". By default "unit" tile is defined in technology file and the other unit tile "lvt_unit" is also added to the technology file. Since we have multi height cells in the reference library separate placement rows have to be provided for two different unit tiles. The core area is divided into two separate unit tile section providing larger area for Hvt unit tile as shown in the Figure 7.

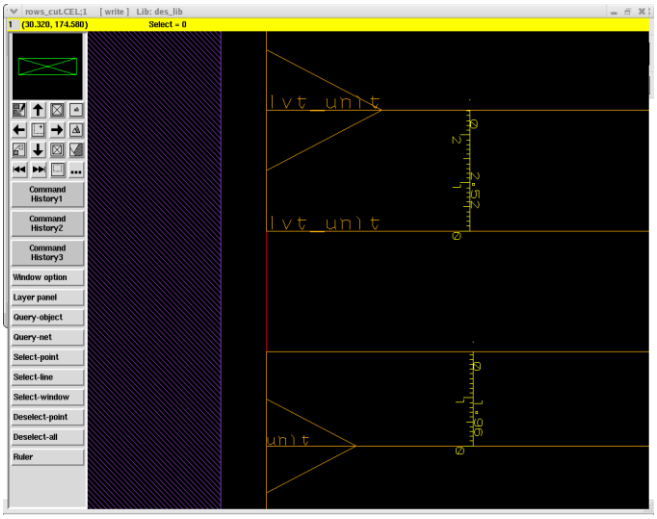

Fig. 7Different unit tile placement

ISSN No:-2456-2165

First as per the default floor planning flow rows are constructed with unit tile. Later rows are deleted from the part of the core area and new rows are inserted with the tile "lvt_unit". Improper allotment of area can give rise to congestion. Some iteration of trial and error experiments were conducted to find best suitable area for two different unit tiles. The "unit" tile covers 44.36% of core area while "lvt_unit" 65.53% of the core area. Legal placement of the standard cells is automatically taken care by Astro tool as two separate placement area is defined for multi heighten cells. But this method of placement generates unacceptable congestion around the junction area of two separate unit tile sections. The congestion map is shown in Figure 8.

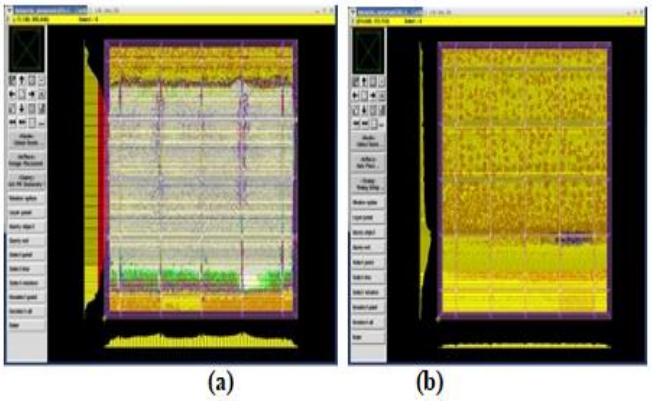

Fig. 8 Congestion: (a) Congestion with aspect ratio 1 and core utilization of 70% (b)Reduced Congestion with specified core height and width set to 950 µm

There are two congestion maps. One is related to the floor planning with aspect ratio 1 and core utilization of 70%. This shows horizontal congestion over the limited value of one all over the core area meaning that design can't be routed at all. Hence core area has to be increased by specifying height and width. The other congestion map is generated with the floor plan wherein core area is set to 950 µm. Here we can observe although congestion has reduced over the core area it is still a concern over the area wherein two different unit tiles me rge as marked by the circle. But design can be routable and can be carried to next stages of place and route flow provided timing is met in subsequent implementation steps.

Tighter timing constraints and more interrelated connections of standard cells aro und the junction area of different unit tiles have lead to more congestion. It is observed that increasing the area isn't a solution to congestion. In addition to congestion, situation verses with the timing optimization effort by the tool. Timing target is not able to meet. Optimization process inserts several buffers around the junction area and some of them are placed illegally due to the lack of placement area. Since the timing is not possible to meet design has to be abandoned from subsequent steps. Hence in a multi vt design flow cell library with multi heights are not preferred.

In this flow first the design is synthesized using Hvt library with the same constraint file provided earlier and later incremental compilation is carried out using Lvt cell library. Both libraries have same cell height of 1.96 µ. Floor planning and power planning are same as in the previous experiments. Multi Vt mode optimization is enabled for Astro timing optimization. There is no congestion and the design is easily routable. After placement there is a setup violation. Since the violation value is less the design is taken to routing stage and high effort optimization fixed most of the violations. For some of the critical paths manual swapping of the high vt cell instances with low vt cells are done.

Both general power model and IR/EM violation analysis are carried out using the tool Astro Rail. Power supply voltage specified for this analysis is: minimum-0.8V, typical-1.2V and maximum-1.32V. As VCD file was not available 0.3 net switching activity is provided in scheme mode. Similar kind of power analysis is carried out for all other Multi Vt flow. Figure 9 presents the IR drop analysis for the design.

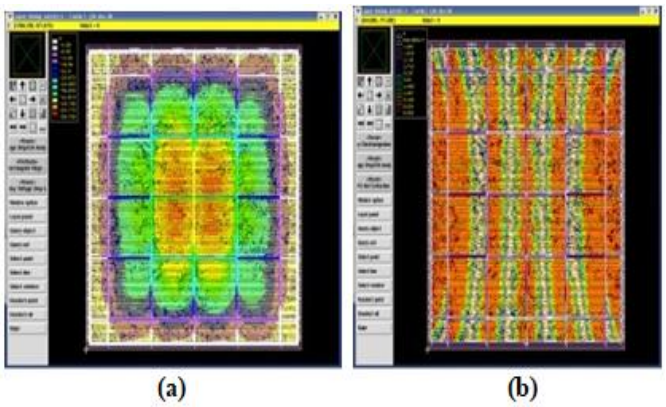

Fig. 9 (a) IR drop map and (b) EM violation map

Results of power analysis for different types of design flows are tabulated in Table 2**.** The results can be analysed based on the leakage power consumption of each design flow as well as number cell count. Design implemented with multi height cells finally abandoned as it could not meet timing targets due to heavy congestion in implementation stage. This design has met the timing targets satisfactorily in the synthesis stage. But note that synthesis engine never bothered about physical information of the standard cells. At the place ment stage since the area is divided into two separate parts to accommodate two different unit tiles there is junction formed around this. Now a timing path has both Lvt and Hvt cell which have to be placed in different core areas and also this placement should be as nearer as possible to reduce net delays. Thus tool tries to place all the cells around junction area hence naturally congestion is created. In addition to this illegal placement of the standard cells are also observed. Power consumption is also more due to the larger height (2.52µm) and hence area of the Lvt cells. Remaining results were obtained with common height cell libraries. In these experiments different Multi Vt design flows are studied to understand the power consumption variations in different situations. It can be observed from the table most of the Multi Vt design flows perform well as per as the leakage power is concerned. Since the timing constraint was tighter for the netlist considered for these

ISSN No:-2456-2165

experiments Lvt cells are more utilized by the tool in all design flows as it is faster. It is straight forward that design implementation using Lvt cells will give +ve slack but leakage power more and Hvt cells will be viceversa. Comparing the leakage value of the regular Vt implementation with the Hvt to Mvt and single pass Mvt flows we can observe Mvt flow has an edge over the other flow as per as theleakage is concerned. Mvt flow improved the leakage by around 15%. Comparison of power results from synthesis and physical implementation reveals that implementation slightly (sometimes large) deteriorates the power efficiency. Increased usage of Low Vt cells at the physical design level increases leakage power. Power analysis in DC by providing parasitic information in the form of SPEF generated from the physical implementation can give more accurate dynamic power analysis result. Area is calculated based on equivalent NAND gate size. As per as the area is concerned Mvt design methodology produces less number gates both in front end and back end design flow compared to the any other flow. Next leas t count is produced by Lvt to Mvt flow. Both Hvt and Hvt to Mvt flow result in large gate count. Thus from the result obtained we can say that either Mvt or Lvt to Mvt flow is best suitable to reduce the area. Since these methodologies also produce less leakage these are most suitable for the designs with tight timing constraints. Note that with the relaxation of timing constraints leakage results alter and other methodology may do well than observed here.

Table 2 Summary of power analysis results obtained from Muti Vt design flows

Netlist Without DFT; multi height cell library					
		Leakage Power	**Cell instances**	**Gate count**	**Timing**
hvt-->Mvt	Synthesis	4.8127	Hvt-11458 Lvt-59321	176947	met
	PhysicalDesign	4.68685	Hvt-4693Lvt-65756	176122	not met
Netlist Without DFT; same height cell library					
		Leakage Power	**Cell instances**	**Gate count**	**Timing**
Normal vt	Synthesis	0.3838	66181	168965	met
	PhysicalDesign	-	-	-	-
hvt-->Mvt	Synthesis	0.7415103	Hvt-13632Lvt-72688	215800	met
	Physical Design	0.821961	Hvt-11038 Lvt-75936	217435	met
Mvt	Synthesis	0.3168684	Hvt-8775 Lvt-53540	151034	met
	Physical Design	0.646354	Hvt-6784 Lvt-65045	179572	met
Lvt	Synthesis	0.4205244	64262	160655	met
	Physical Design	0.682563	68256	170640	met
Lvt->Mvt	Synthesis	0.3015456	Hvt-17071Lvt-46641	150956	met
	PhysicalDesign	0.382589	Hvt-14017Lvt-48211	147439	met
Hvt	Synthesis	0.0270901	70843	215706	-4.57 ns
	PhysicalDesign	-	-	-	-
Netlist With DFT; same height cell library					
		Leakage Power	**Cell instances**	**Gate count**	**Timing**
hvt-->Mvt	Synthesis	0.7139687	Hvt-10550 Lvt-72304	207135	met
	PhysicalDesign	0.738956	Hvt-10142Lvt-78639	221952	met

VI. CONCLUSION

In this paper modified adaptive Viterbi decoder architecture with message decoding logic and modified adaptive decoder architecture is presented. 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. Multi Vt design flow with multi height cell library resulted in a slack of -1.4 ns in placement stage and hence can't be implemented. Mvt design flow produces a leakage of 0.3168mW at synthesis level which is less compared to the Rvt flow and a leakage power of 0.6463mW at the physical implementation level which is more than the Rvt flow. Lvt to Mvt flow resulted in less leakage power of 0.301mW and 0.383mW in synthesis and physical implementation respectively. The same design methodology provides least area of 150.9K and 147.4K in synthesis and physical implementation respectively. Hvt design me thodology even though gives least leakage power of 0.027mW; it does not meet timing by -4.57ns at synthesis level. Hence Hvt design methodology can't be used for leakage optimization. At synthesis stage of the design around 9% of leakage power reduction is observed. After physical implementation, leakage power increased by 25%. This variation is attributed to the pessimistic timing constraints. Special issues like multi height cells, floor planning blockages and restrictions, tighter timing constraints may hinder the advantages of Multi Vt implementation in physical design. At the worst case the design may not be implementable. Multi Vt design flow is not recommended for design with tight timing and leakage power targets. For such kind of design requirements other low power approaches such as power gating, clock gating should be examined. Even though there is large improvement in the leakage power reduction from regular Vt flow to Multi Vt flows in synthesis level, it may improve or degrade in physical implementation based on optimization strategy, constraints and design complexity.

REFERENCES

[1]. Mandar Chitre, Shiraz Shahabudeen, Lee Freitag, Milica Stojanovic, Recent Advances in Underwater Acoustic Communications & Networking,

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

[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), June 2013, pp. 1–5

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

[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, August 2016, UNIVERSITY OF THE RYUKYUS, http://ir.lib.u-ryukyu.ac.jp/bitstream/20.500.12000/35705/3/rikoken324text.pdf

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

[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, July 2009

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

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

[13]. Jie Jin & Chi- ying Tsui 2007, '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.

[14]. Jinjin He, Huaping Liu, Zhongfeng Wang, Xinming Huang & Kai Zhang 2012, '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.

[15]. Yan Sun & Zhizhong Ding 2012, 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.

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

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

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

[19]. Yao Gang &Tughrul Arslan 2004, '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.
[20]. Dalia A. El-Dib & Mohamed I. Elmasry 2004, '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.
[21]. Man Guo, Omair Ahmad, M, Swamy, MNS & Chunyan Wang 2005, '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.