Macrothink Institute™

Network Protocols and Algorithms
ISSN 1943-3581
2014, Vol. 6, No. 1

Speed, Noise Immunity, Power Consumption and Area Comparison between Different Approaches of Low-Power Viterbi Decoder for Digital Wireless Communication Applications

Sayak Bhowal

Advanced Signal Processing Group, Centre for Development of Advanced Computing
(CDAC), Kolkata

Plot- E-2/1, Block-GP, Sector - V, Saltlake Electronics Complex, Bidhannagar,
Kolkata - 700 091

Mobile: +919748638176 E-mail: sayak.bhowal@cdac.in

Received: December 15, 2013 Accepted: March 10, 2014 Published: April 30, 2014

DOI: 10.5296/npa.v6i1.4759 URL: http://dx.doi.org/10.5296/npa.v6i1.4759

Abstract

Noise immunity and speed are two vital issues for designing encoding-decoding system for wireless communication. Convolutional coding is widely used in wireless communication system for its error correction property. For the decoding purpose of Convolutional coding Viterbi decoder is used. Core module of Viterbi decoder is Adder-Comparator-Selector (ACS) which takes approximately 70% of total power consumption. So, Adder-Comparator-Selector (ACS) module is transformed into Comparator-Selector-Adder CSA) module for power saving. Reduction of Hamming Distance Logic Circuitry for branch metric calculation also saving power but enhances the packing density of the circuit. In this paper the comparison between ACS and CSA is not only described in terms of power reduction and area but also speed and noise immunity are compared. Basically there are three types of Viterbi decoders: namely Register Exchange, Shift Update and Selective Update. These decoders do not follows the parallelism and pipelining concept but folding cascaded designing of Viterbi Decoder supports parallelism which enhance the speed of the system. This paper gives a new idea of logic reduction of Viterbi Decoder as well as comparison of different Viterbi decoders in different aspects.

Keywords: ACS, Convolutional code, CSA, Maximum likelihood (ML) algorithm, Trace back, Trellis tree, Viterbi decoder

1. Introduction

As an extension of the previous paper [1], in this paper the main discussion point is speed, noise immunity and types of Viterbi decoders. In the paper [1], comparison makes through only Adder-Comparator-Selector (ACS) and Comparator-Selector-Adder (CSA) modules of the Viterbi decoder, but whole system comparison is not described. Here each module of Viterbi decoder is compared for different architectures.

Power consumption and area reduction are main concern in VLSI design but for communication approach speed and noise are two factors for designing a system. Mainly Convolutional codes are used in satellite and space communication system due to its error correction capability having memory. The encoder output of Convolutional coding is not only depends on present input but also depends on previous input which enhances the capability of error correction by assumption of present output using previous inputs. This characteristics differs Convolutional coding from Block coding.

The content of the paper is arranged as follows. Section II presents the motivation of implementing new CSA module design instead of ACS module at Viterbi decoder and importance of comparison between different Viterbi decoders. Section III describes the basic principle of Viterbi decoder. Section IV presents the working principle of ACS module and also describes the branch metric generation in conventional way as a related topic. Section V proposes generation of branch metric using new way which is one of the points of focus of this paper. Section VI proposes the transformation of ACS module to CSA module by altering the logic. Different types of Viterbi decode and their technical differences are precisely described at Section VII. Error rate comparison with the help of probabilistic approach and theoretical background is explained under Section VIII. Section IX elaborates the testing results for applying new proposed CSA module in different aspects and compares it by using different Viterbi decoders as a proof of concept.

2. Motivation

Viterbi decoding algorithm is a well known algorithm for network based engineers. It is used for Convolutional code decoding. Various approaches of this algorithm are available in the market. But it is confusing that which approach is suitable in which application. From this background the paper is motivated to compare different approaches. Technically to install a sophisticated system four types of constraints are arises that are namely speed, space, noise immunity and power.

This paper does not confined only on comparison between different approaches of Viterbi algorithms. A new concept of logic reduction is also the motivated area of this paper. Most of the cases it is seen that power consumption makes great issue to install a system. This paper contributes to reduce the power consumption using logic reduction and transfer the Adder-Comparator-Selector block (ACS) to Comparator-Selector-Adder (CSA) block. But to reduce the power consumption space occupancy of the system is increased.

Logically each approach can highly efficient to decode the Convolutional codes. But in presence of noise in communication channel it is not desired that Viterbi decoder gives appropriate result. It is observed that the noise immunity characteristics of different approaches are different. From that background the paper is motivated to explain the probability of error mathematically for different approaches and also prove it practically by simulated result using artificially random noise generator.

3. Working Principles of Viterbi Decoder

Basic building block of Viterbi Decoder is shown in Figure 1. Conceptually there are eight modules which are taken as consideration to make Viterbi Decoder. Role of each modules are described shortly below,

- Input and output Interface Block: These blocks provide the interfaces between external components of the system in serial or parallel form. Most of the cases serial input is coming from the channel and the generated output is produced by parallel form using serial to parallel conversation.

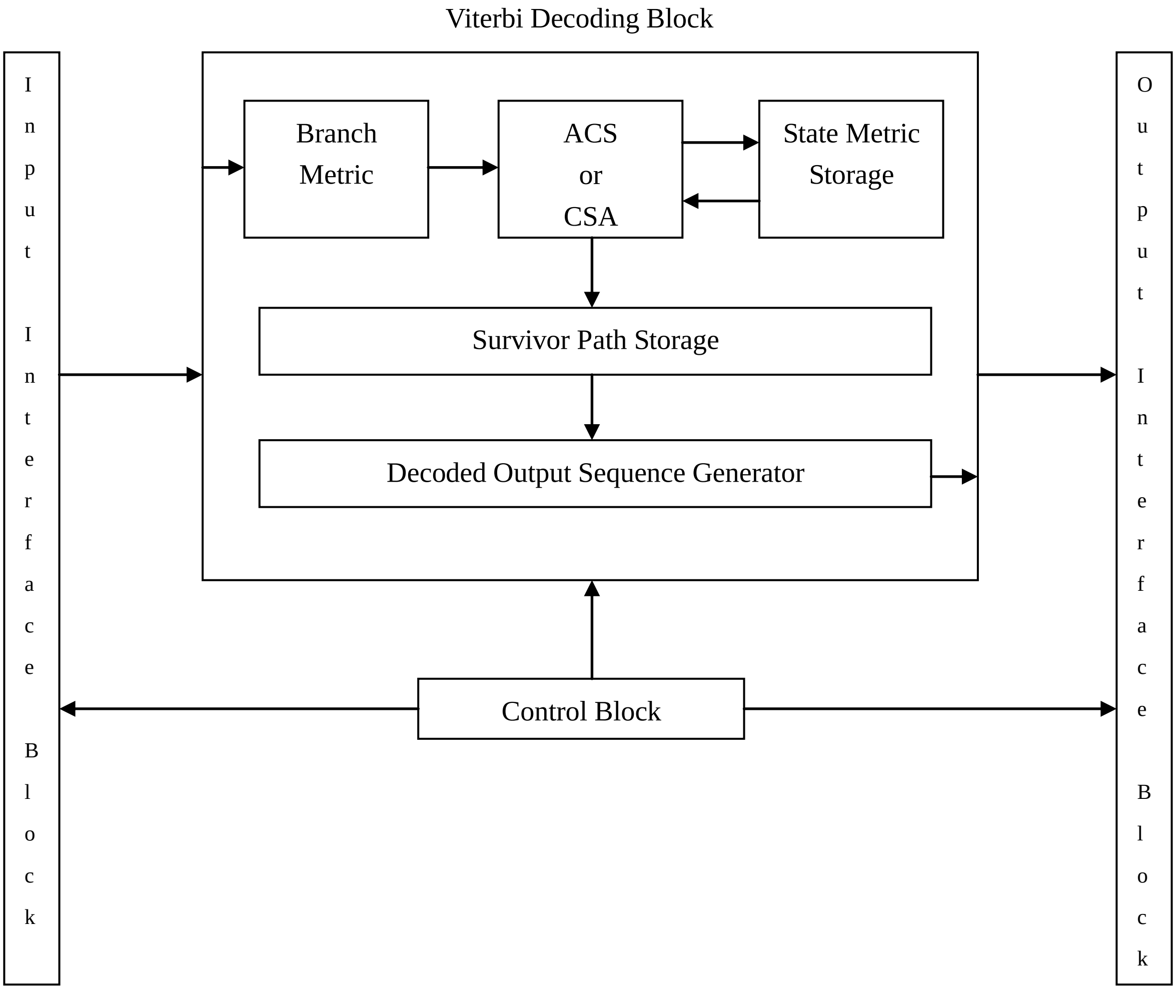

Figure 1. Block Diagram of Viterbi Decoder [2]

- Branch metric: This module calculates the Hamming distance between received codeword and expected codeword which is considered as branch metric value.
- Storage Path Metric: This block stores partial path metric for each stage at current state.
- ACS or CSA: This is the most vital block of Viterbi algorithm using Adder-Comparator-Selector (ACS) or Comparator-Selector-Adder (CSA) logic next stage survivor path is calculated. Maximum Likelihood algorithm is applied for taking right decision. The maximum likelihood (ML) estimation that maximizes the probability *p(r|e)* is incorporated into the Viterbi decoding process [5].
- Survivor Path Storage: This block records the survivor path and each node of trellis tree. It is necessary in trace back approach for error correction.
- Decoded Output Sequence Generator: This block presents for preparing output using combinational logics and reading the memory of Survival Path Storage Block.

4. Working Principle of ACS Module in Viterbi Decoder

The essence of maximum likelihood (ML) decoding is incorporated into the coding of butterfly model to reduce the complexity of computing [5], [2]. The received codeword is compared to the expected codeword and number of differing bits is counted at the branch metric computation block. In Figure 2 the implementation of the block is shown.

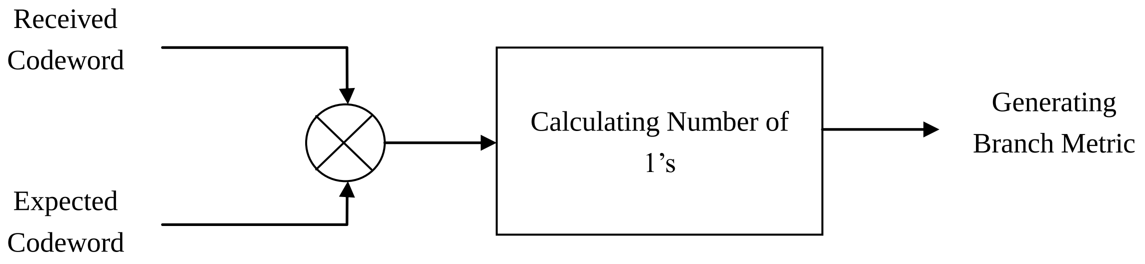

Figure 2. Block diagram of Branch Metric generation

The survivor path is selected by the path metric update block. The butterfly structure is incorporated into the trellis diagram of Convolutional encoder for a rate 1/n [2], [5]. The structure builds up a pair of origin and destination states, which are cross coupled by four interconnecting branches shown in Figure 3.

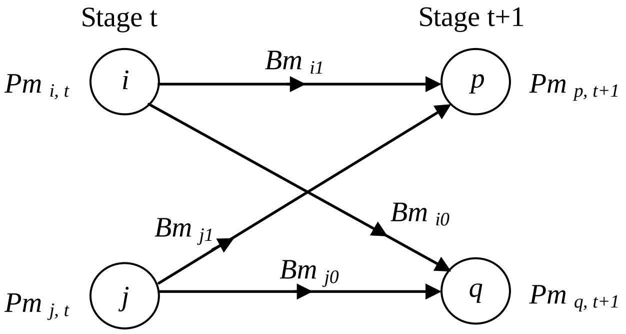

Figure 3. A butterfly structure for a Viterbi Decoder [2]

In the Figure 3, When the corresponding input bit is ‘0’ (‘1’), the lower (upper) branch is selected from each state *i* or *j* to transfer next state. The relationships between the path metric (*Pm*) and the branch metric (*Bm*) is shown below which set up for a Convolutional encoder (n, 1, m) where n is the number of outputs, m is the number of memory elements used in encoder and number of input bit is 1 for each cycle [2], [5].

$$Pm_{p,t+1} = Min[(Pm_{i,t}+Bm_{i1}),(Pm_{j,t}+Bm_{j1})] \tag{1}$$

$$Pm_{q,t+1} = Min[(Pm_{i,t}+Bm_{i0}),(Pm_{j,t}+Bm_{j0})] \tag{2}$$

Where,

$$p=2i \tag{3}$$

and

$$q=2i+1 \tag{4}$$

From (3) and (4) the point to be noted that state *p* is always even (*p* = *2i*) and state *q* is always odd (*q* = *2i+1*). It signifies that an even (odd) state is accounted only if the input bit is ‘1’ (‘0’). From this above property the trace back concept of decoding is evolved. So, it is possible to go back from a state at stage t+1 to a state at previous stage t on condition of survivor path of that state is known. As an example if the survivor branch is lower (upper) path for an odd state *q* at stage t+1, then the previous state at stage t will be state *j (i)*. The trace back procedure is applied at state *p* in similar way. In brief, if the survivor path is recorded the system can trace back from final state to initial state by following above technique.

The Add-Compare-Select (ACS) module invokes the butterfly structure [5], [2]. An ACS module for state *p* in Figure 3 is shown in Figure 4 and state *q* is shown in Figure 5. The partial path metric is calculated by adding of previous state path metric and branch metric of upper (lower) branch [7]. The comparator compares these two partial path metrics and selector selects the minimum partial path depends on comparator decision [7]. The selector output updates the path metric of state *p (q)*.

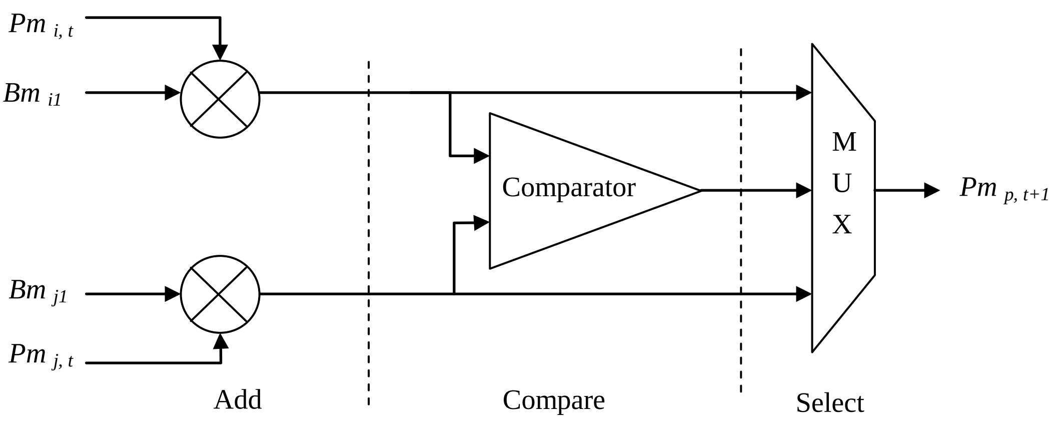

Figure 4. ACS module for State *p* [5]

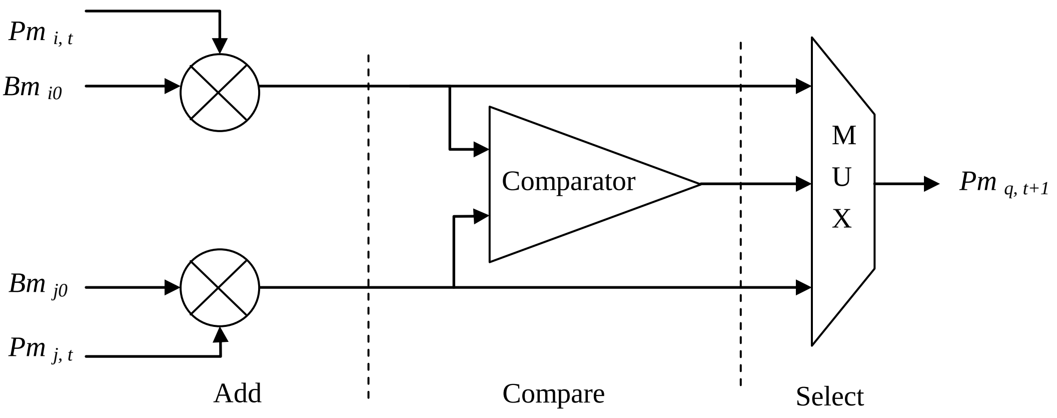

Figure 5. ACS module for state *q* [5]

The number of necessary ACS module for Viterbi decoding is depends on number of total states. Generally it is half the number of total states [5]. For long chain input sequence same ACS circuit is used as time sharing mode for parallel architecture, but it hampers the speed of operation due to context switching.

4.1 Generation of branch metric in conventional way

The branch metric computation block compares the received codeword with the expected codeword and counts the number of differing bits which is called Hamming distance measurement. To measure Hamming distance first XOR operation is performed by received codeword and expected codeword, then the result goes to 8x3 Decoder for getting the number of 1's differs from received codeword and expected codeword. An implementation of the block is shown in Figure 6.

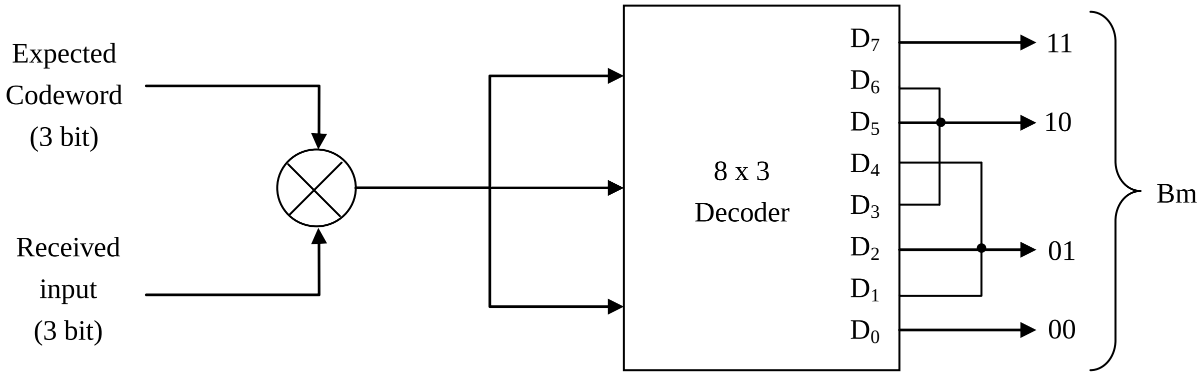

Figure 6. Hamming distance calculation block in conventional way

As an example, taking expected codeword is 010 so the truth table of above logic is given below at Table 1,

Table 1. Truth table of the circuit of Figure 6

Received Codeword	Expected Codeword	XOR Output	8x3 Decoder Output	Branch Metric
000	010	010	D_2	01
001	010	011	D_3	10
010	010	000	D_0	00
011	010	001	D_1	01
100	010	110	D_6	10
101	010	111	D_7	11
110	010	100	D_4	01
111	010	101	D_5	10

5. Generation of branch metrics in new architecture

The new trend of VLSI is to reduce the power keeping the same logic output. In this section, the changing of previous section circuit for power saving by keeping the same output logic is discussed. To generate the Branch metric in new architecture we first assume that the designer should know the expected codeword from the Trellis Tree of a specific Viterbi decoder. As the designer previously knows the expected codeword he/she can minimize the logic of branch metric generation by using a simple Full Adder and using a dedicated hardware for each branch metric. An implementation of the block is shown in Figure 7. In the block diagram ≀≀ symbol is given for indicating that NOT gates included depending on expected codeword.

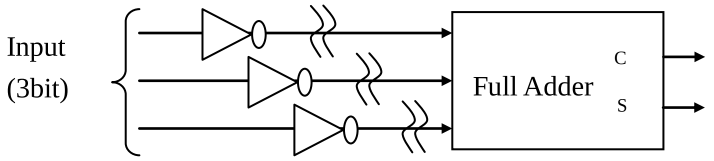

Figure 7. Generalized Hamming distance calculation block in proposed design

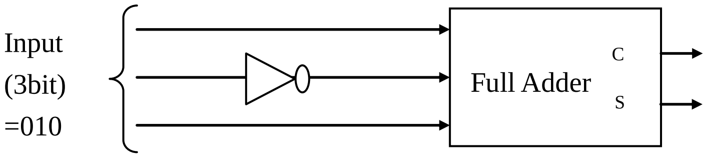

Figure 8. Hamming distance calculation block in proposed design for expected codeword 010

Taking the same example, the expected codeword is 010, so the truth table of Figure 8 is given below at Table 2,

Table 2. Truth table of the circuit of Figure 8

Received Codeword	Full Adder Input	Full Adder Output		Branch Metric
		Carry	*Sum*	
000	010	0	1	01
001	011	1	0	10
010	000	0	0	00
011	001	0	1	01
100	110	1	0	10
101	111	1	1	11
110	100	0	1	01
111	101	1	0	10

6. Transformation of ACS unit to CSA unit in new architecture

From Figure 4 and Figure 5, it can be concluded that four times add operations, twice compare operations and twice select operations are needed for each ACS operation. A large number of operations result in high power consumption and large delay, which are not suitable for high-speed low-power applications.

Now, we can first compare the two path matrices to find which value is lower. From this comparison the system can decide the lower valued path metric and this lower valued path metric is added with two different branch metrics to get new path metrics for next iteration. There is no need to adding two path metrics with branch metrics as on ACS because one branch metric is always in inverted form of other branch metric and also the values of branch metrics are very much lower than path metrics. So, there is no impact of branch metric values in comparison. Only path metric values take the major role of comparison.

There is another aspect of study in Viterbi Decoder that we know the number of Flip-Flops used in encoder is less than or equal to Number of encrypted bits +1. So, it is desirable that for serial communication after (Number of encrypted bits +1)th counting no received codeword is found for each iteration. But if the noise arise into the channel then we found that the received codeword also come after (Number of encrypted bits +1)th counting.

From the above logical sense we compare counter output with a fixed number whose value depends on encoder design. This fixed number is actually the number of input bits generated by Convolutional encoder plus one. Now we take high (‘1’) output of comparator which is connected with counter for counting less than equal to number of flip-flops or number of input bits +1. If AND operation is performed between Comparator output and Path Metric comparison output then the output of AND operation used as a selection path of the multiplexer of path matrices.

Following block diagram in Figure 9 is depicted for new CSA architecture which takes only two 6 bit adders, compromising one extra multiplexer (in conventional way two multiplexers are needed for selection block) compared with conventional ACS unit.

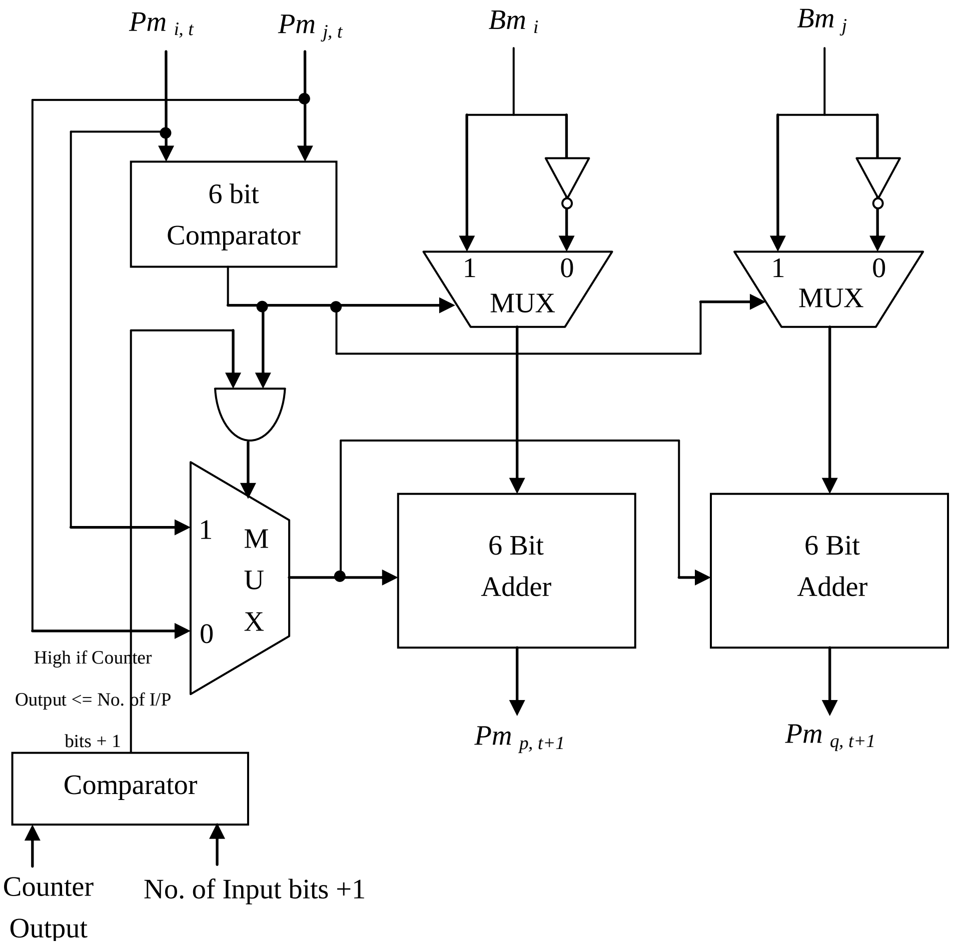

Figure 9. CSA module in proposed design

7. Types of Viterbi Decoder

Basically Viterbi Decoders follow generally two approaches, one is trace back and another is without trace back [6]. In without trace back approach huge memory register is needed for storing the possible partial outputs of each stage from initial stage to final stage. As a result it gives fine result in noisy environment. It is quite natural that as partial outputs are stored step by step, without trace back approach is more noise immune but it is time consuming for storage and also it takes more area for memory element. Without trace back approach is called in another name register exchange approach as huge registers are needed for design the system [6].

But for trace back approaches, each stage takes hard decision and don't need to store the partial output of each stage always. Because the node entry can be stored by which the control can trace back from final stage to initial stage and generates output [6]. There are two types of methods incorporate for trace back approach namely, Selective Update and Shift Update.

In selective update method each selected node value is stored by which trace back follows the path from final stage to initial stage and recover the actual decoded output. But in shift update method parallel paths of nodes are stored in different registers and finally minimum Hamming distance calculated path is selected as an actual path. It gives nearly same noise less output like register exchange approach but it does not take huge memory for storage. Shift update method is too critical than other methods.

In modern technology parallelism offers enhancement of speed. Parallelism can be incorporated into Viterbi Decoder by using Folding Cascade method [8]. For parallelism serial input is needed to make parallel and total system can be by parts into small subsystems which takes input and generates partial outputs. From each partial output actual decoded output is calculated using intelligence. This approach gives highly appreciative speed into the system but synchronization is necessary for getting good result from this type of system.

8. Comparison of Error rates in Different Viterbi Decoding Approaches

From the previous analysis it is clear that for register exchange approach error rate is lowest, but others have little bit greater error rates than register exchange approach.

Let us, consider that the bit duration of a codeword is T_c and the bit duration of decoded word is T_d. Now, for Convolutional coding T_d is obviously greater than T_c because for one bit decoded word several numbers of coded words are required.

Now, if we consider the signal power P_s, thermal noise power spectral density η and number of bits in a word n for each system is equal, then the probability of error in each bit is equal to,

$$P = \frac{1}{2} erfc\sqrt{\frac{P_s T_d}{n\eta}} \tag{5}$$

For register exchange approach the decoded codeword will be erroneous if one or more digits in coded word are in error. So, the probability of digits which are not erroneous is $1\text{-}P$. Then the probability for all n digits are not in error is $(1\text{-}P)^n$. So, the probability for at least one bit error is

$$P_e = 1-(1-P)^n \cong nP \tag{6}$$

For Selective update approach using *(n,k)* error correcting Hamming Distance. Then a error occurs in decoded codeword only if two or more than two error occurs. So, the probability for at least one bit error is

$$P_e^{selective} = \binom{n}{k}(1-P)^{n-k}P^k \tag{7}$$

Where $\binom{n}{k}$ is the number of combination of *n* where *k* thinks are at a time from *n*.

For shift update approach also using *(n,k)* error correcting Hamming Distance but it records node values for all possible combinations from initial stage to final stage. So, the probability for at least one bit error is

$$P_e^{shift} = \frac{k}{n}\binom{n}{k}(1-P)^{n-k}P^k \tag{8}$$

For folded cascade approach if the system is broken up m number of subsystem then each system is highly cohesive with each other. So, the probability for at least one bit error is

$$P_e^{folded} = m\binom{n}{k}(1-P)^{n-k}P^k \tag{9}$$

9. Test Results and Analysis

The area and power dissipation is measured in three different approaches of Viterbi decoders: the register-exchange scheme, shift update scheme and selective update scheme [6].

Three different approaches of Viterbi decoders are implemented in the standard cell environment. First, these three Viterbi decoders are developed at the register transfer level in VHDL and synthesized them using a Synopsys tool [6]. Six metal layer 0.25 µm technologies with the supply voltage of 1.8V is used to develop the whole system using ACS module as well as CSA module. Power dissipation and also the area management are estimated for the synthesized gate-level circuits using Synopsys tools [6].

To compare the result first the analyzing report of those three decoder are collected in convention ACS designing way. Then CSA module is implemented by replacing the ACS module. After that the analyzing report is collected again. The new designed module reduces nearly 10% power dissipation in each scheme whereas it takes more area for implementation. The reports are given below at Table 3 and Table 4,

Table 3. Testing report of three approaches in conventional way

Module	Register Exchange approach		Shift Update approach		Selective Update approach	
	Power	*Area*	*Power*	*Area*	*Power*	*Area*
Adder-Comparator-Selector Block	818.5 µw	57584 µm^2	818.5 µw	57584 µm^2	818.5 µw	57584 µm^2
Decoded output sequence generator	152.2 µw	110507 µm^2	0 µw	0 µm^2	0 µw	0 µm^2
Shift Register	125.7 µw	4770 µm^2	125.2 µw	4765 µm^2	125.2 µw	4765 µm^2
Survivor path storage and traceback	0 µw	0 µm^2	576.1 µw	91739 µm^2	189.8 µw	87985 µm^2
Counter	72.9 µw	1521 µm^2	72.9 µw	1521 µm^2	72.9 µw	1521 µm^2
Total	1169.3 µw	174382 µm^2	1592.7 µw	155609 µm^2	1206.4 µw	151855 µm^2

Table 4. Testing report of three approaches in proposed design

Module	Register Exchange approach		Shift Update approach		Selective Update approach	
	Power	*Area*	*Power*	*Area*	*Power*	*Area*
Comparator-Selector-Adder Block	704.0 µw	109892 µm^2	710.0 µw	104270 µm^2	606.6 µw	103141 µm^2
Decoded output sequence generator	152.2 µw	110507 µm^2	0 µw	0 µm^2	0 µw	0 µm^2
Shift Register	125.7 µw	4770 µm^2	125.2 µw	4765 µm^2	125.2 µw	4765 µm^2
Survivor path storage and traceback	0 µw	0 µm^2	576.1 µw	91739 µm^2	189.8 µw	87985 µm^2
Counter	72.9 µw	1521 µm^2	72.9 µw	1521 µm^2	72.9 µw	1521 µm^2
Total	1054.8 µw	226690 µm^2	1484.2 µw	202295 µm^2	994.5 µw	197412 µm^2

After comparing Table 3 and Table 4 it can be decided that power consumption of CSA module reduces overall power consumption compared with ACS module and CSA module does not affects other modules. Besides as a drawback, area coverage of CSA module is higher than ACS module.

Using the data of Table 3 and Table 4, Figure 10, 11, 12 and 13 are depicted as graphical representation of the same for better visualization.

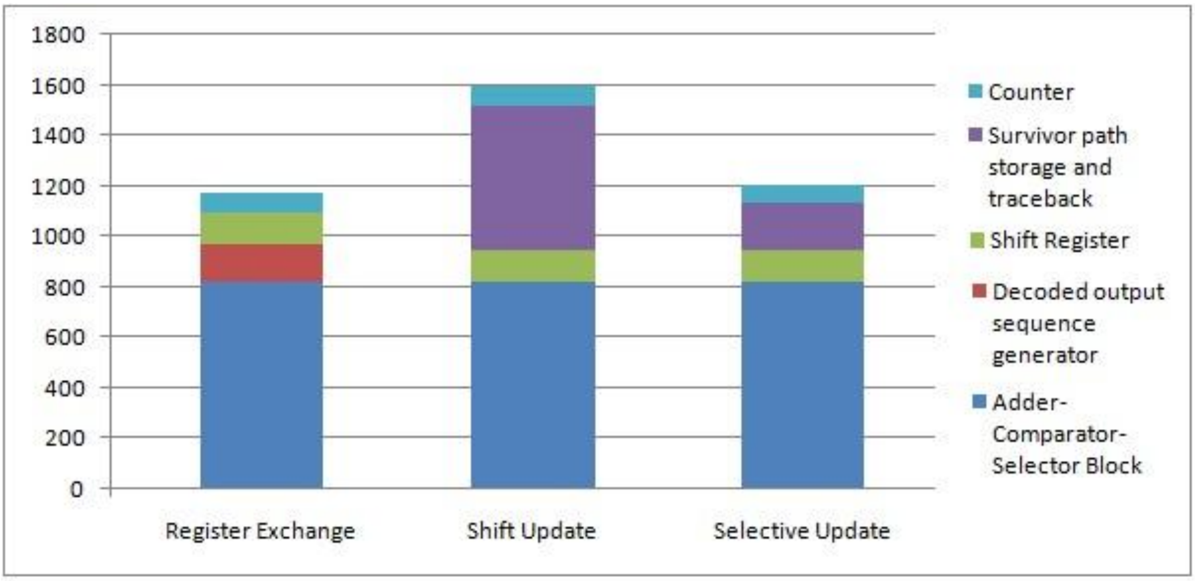

Figure 10. Graphical Representation of Power Consumption Testing Report of three approaches in conventional way

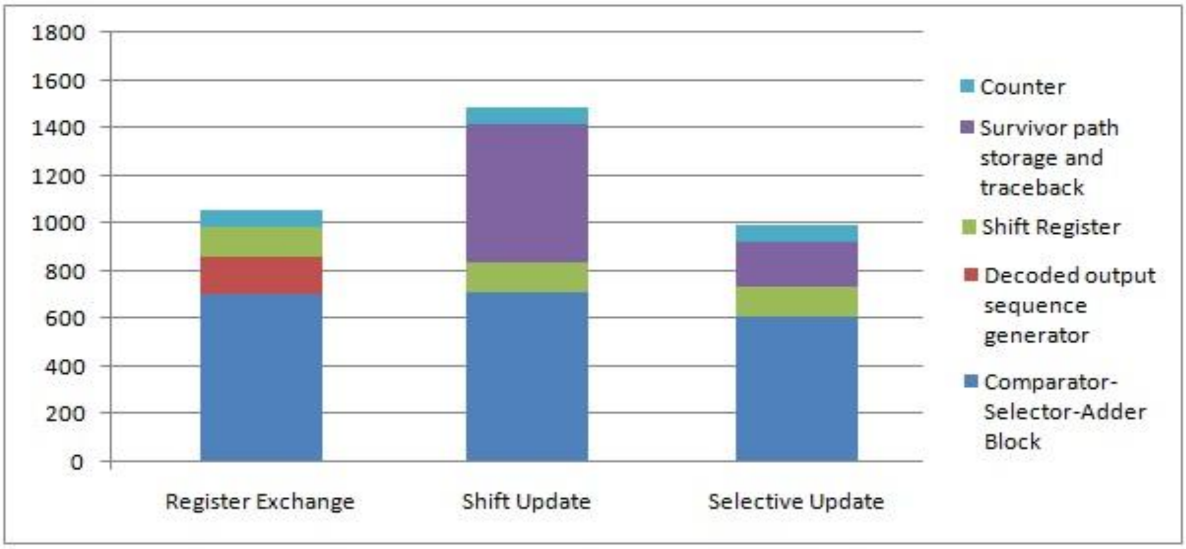

Figure 11. Graphical Representation of Power Consumption Testing Report of three approaches in proposed design

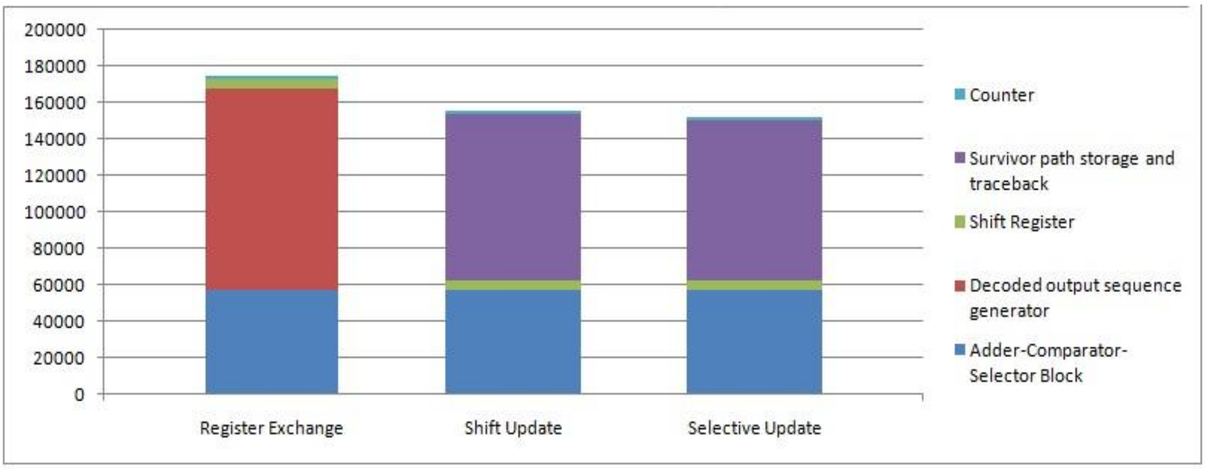

Figure 12. Graphical Representation of Coverage Area Testing Report of three approaches in conventional way

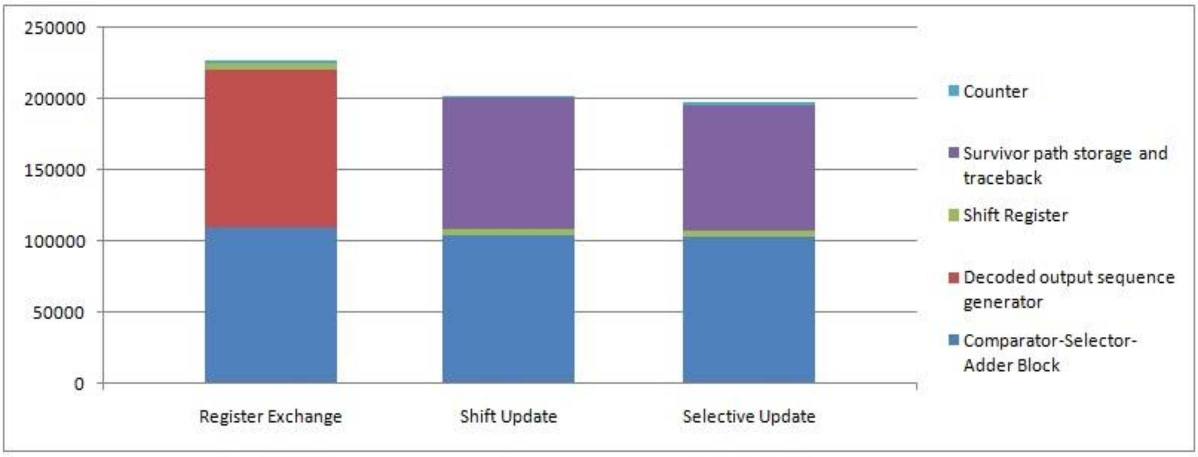

Figure 13. Graphical Representation of Coverage Area Testing Report of three approaches in proposed design

From the testing report it is found that the power dissipation is reduced nearly 10% for each approaches whereas the total area of the circuit is increased. The total area is increased due to the cause of dedicated full adders are used in Hamming distance calculation for each branch metric generation. So, the total area depends on the number of branch metric in a circuit. In case of branch metric generation in conventional way only one circuit (shown in Figure 6) is sufficient for any branch metric generation.

9.1 Noise Immunity Test Result

For test purpose of noise immunity an artificial random noise generator is designed using VHDL Code, which inverts an input bit randomly and send it to the decoder. Now depending of randomness and noise frequency the result may vary but the probability factors which are already described are not changed. For experimental purpose 50 byte input stream is sent into each Viterbi system and noise generator tampers several input bits randomly. After comparing with output sequence following graph is generated.

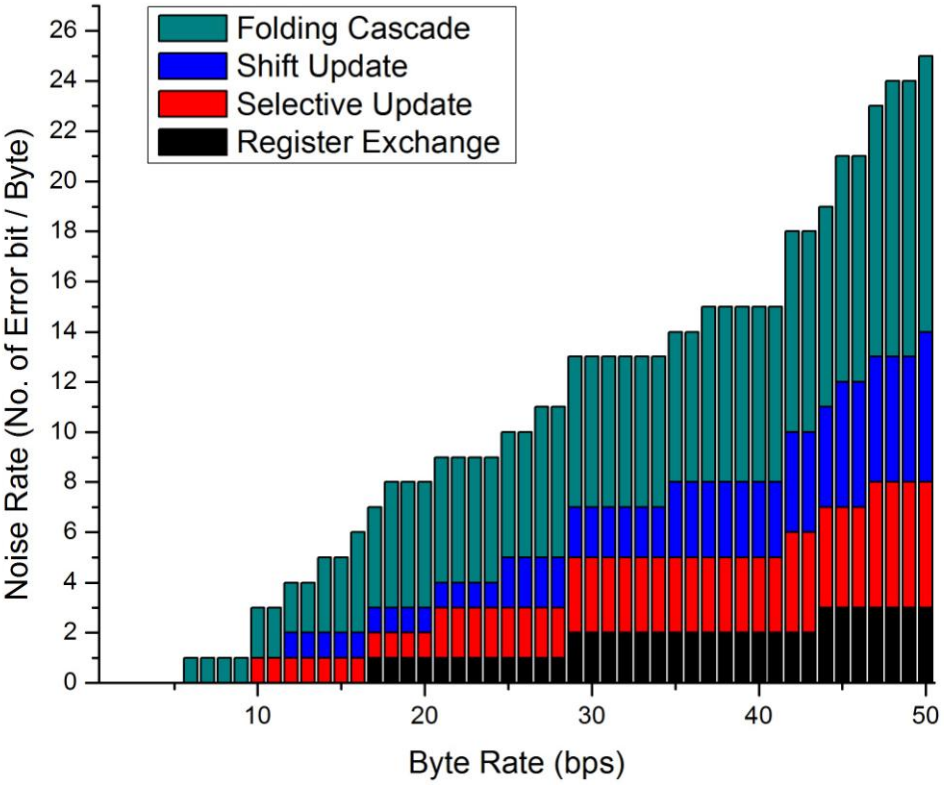

Figure 14. Noise Rate Vs Byte Rate Graph

From Figure 14 it is clear that for Register Exchange approach low number of error bits is generated at the output sequence where as for folding cascade approach high number of error bits is present at the output sequence. For Shift update and selective update approach error bit rates of output sequence in between them. This result happens because of random noise generator generates noise in high frequency which completely changes the input bit patterns. As a result trace back path of the decoded system is hampered and generates bad result. In practical scenario this type of high frequent noise is not expected into the communication channel. Then all type of Viterbi decoder gives satisfactory result.

It should be noted that as logic reduction is taking place to reduce power without changing output in proposed design, the new circuit behaves nearly same in noise immune characteristics. As same result is found in both cases taking same noisy encoded signal as input, it is not shown here.

9.2 Speed Test Result

For testing the speed of different Viterbi decoder, a very simple experiment can be done. For any instance of input signal a system lagging is incorporated with the output sequence for calculation of tracing path into the decoder. As a result the output sequence comes several times after. This lagging period varies for different Viterbi systems.

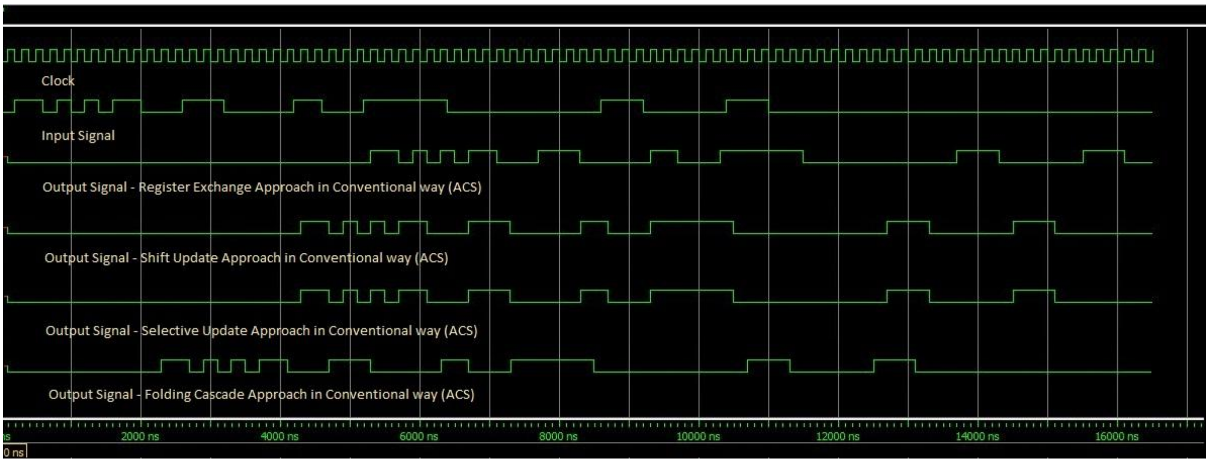
Clock
Input Signal
Output Signal - Register Exchange Approach in Conventional way (ACS)
Output Signal - Shift Update Approach in Conventional way (ACS)
Output Signal - Selective Update Approach in Conventional way (ACS)
Output Signal - Folding Cascade Approach in Conventional way (ACS)
2000 ns
4000 ns
6000 ns
8000 ns
10000 ns
12000 ns
14000 ns
16000 ns
0 ns

Figure 15. Input Signal and Different approaches Output Signal using conventional way of ACS design

From Figure 15, it can be said that Register Exchange approach takes long time to generate output signal corresponding input. Whereas Folding Cascade method takes most short time to generate output signal. Shift Update and Selective Update both takes same time which is faster than Register Exchange but slower than Folding Cascade.

For the proposed design of logic reduction for Hamming distance calculation is able to reduce the lagging time at receiver end because only Full Adder is used for calculation of Hamming distance in proposed design whereas an extra 3x8 Decoder is used in conventional design of Hamming distance calculation. The result of Input Signal Vs Output signal for proposed design is given below at Figure 16.

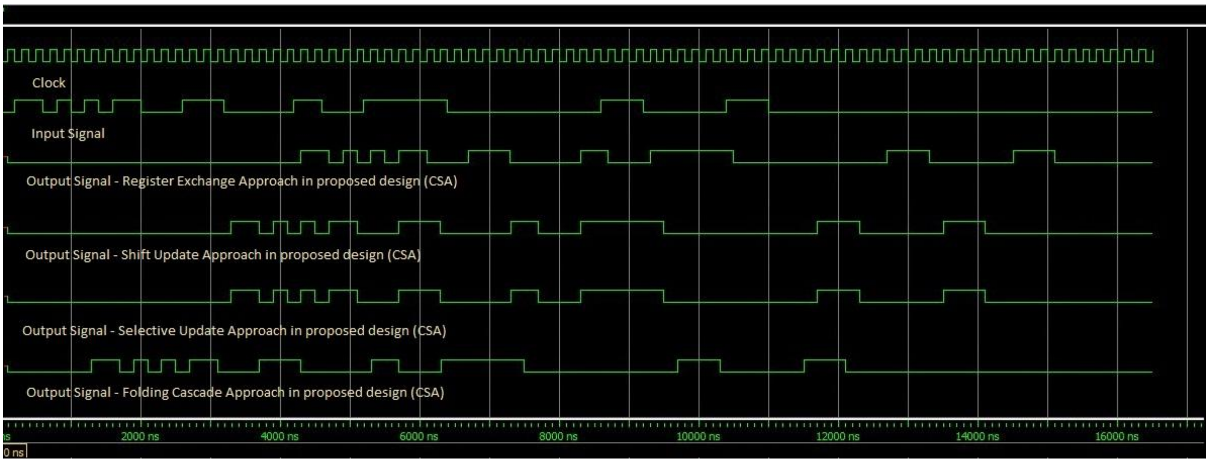
Clock
Input Signal
Output Signal - Register Exchange Approach in proposed design (CSA)
Output Signal - Shift Update Approach in proposed design (CSA)
Output Signal - Selective Update Approach in proposed design (CSA)
Output Signal - Folding Cascade Approach in proposed design (CSA)
2000 ns
4000 ns
6000 ns
8000 ns
10000 ns
12000 ns
14000 ns
16000 ns
0 ns

Figure 16. Input Signal and Different approaches Output Signal using proposed way of CSA design

10. Conclusion

This paper proposes a CSA module implementation which optimizes Viterbi decoder circuits. Here reduced circuit Hamming distance calculation and first comparison method for path metric calculation at CSA module improves the power consumption up to 10% compromising the total area of the circuit. The counter output is compared with number of input bits for error minimization in CSA module. Experimental results indicate that the proposed CSA module decreases nearly 10% of power consumption. This proposed CSA module is applicable in different approaches of Viterbi decoder.

In case of noise immunity comparison it can be concluded that noise immunity of Viterbi decoder depends on the formation and well structured trace back path. But trace back path itself takes complex circuitry as well as area and time consumed. But well structured trace back path gives nearly accurate result theoretically. Register exchange approach does not take the advancement of trace back approach and stores all the possible paths using large number of registers and performs as a best noise immune Viterbi Decoder. Other Viterbi decoders takes the advancement of Trace Back approach which is faster, but due to worst noisy environment trace back approach does not follow the proper path. Hence generates noisy outputs. Folding Cascade method not only takes the advancement of trace back approach but also takes the advancement of parallelism. But the main reason of noisy output for Folding Cascade method is that when an error bit is passed through the system the synchronization of parallelism is going to break. As a result erroneous output is generated until it recovers at least one byte of correct output.

In case of speed comparison it can be concluded that the speed of Viterbi decoding is enhanced by parallelism of architecture. Folding Cascading approach performs as fully parallelized manner and takes very small time to generate output. The lagging period between output and input sequence signifies the speed of system.

References

[1] Bhowal, S. “Transformation of ACS module to CSA module of low-power Viterbi decoder for digital wireless communication applications,” International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013, http://dx.doi.org/10.1109/ICACCI.2013.6637182

[2] Guan, Xuguang , Zhou, Duan ; Wang, Dan, Yang, Yin-tang, Zhu, Zhang-ming, "A Case Study on Fully Asynchronous ACS Module of Low-power Viterbi Decoder for Digital Wireless Communication Applications," International Conference on Computational Intelligence and Natural Computing, 2009. CINC '09. Volume:2, http://dx.doi.org/10.1109/CINC.2009.64

[3] A. J. Viterbi, “Error bounds for Convolutional coding and an asymptotically optimum decoding algorithm,” IEEE Transactions on Information Theory, 1967, http://dx.doi.org/10.1109/TIT.1967.1054010

[4] A. J. Viterbi, “Conventional codes and their performance in communication systems,” IEEE Transactions on Communication, vol.COM-19, 1971, pp. 751-772

[5] In-kyu Lee, Jeff L. Sonntag, “A new Architecture for the Fast Viterbi Algorithm,” IEEE Transactions on Communication, vol. 51, 2003, http://dx.doi.org/10.1109/TCOMM.2003.818100

[6] Ranpara, Samirkumar, Ha, Dong-Sam Sam, "A low-power Viterbi decoder design for wireless communications applications," Twelfth Annual IEEE International ASIC/SOC Conference Proceedings 1999. http://dx.doi.org/10.1109/ASIC.1999.806538

[7] Fei Sun, Tong Zhang, “Low power state-parallel relaxed adaptive Viterbi decoder design and implementation,” IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006. Proceedings, 2006, http://dx.doi.org/10.1109/ISCAS.2006.1693707

[8] Hsiang-Ling Li, Chakrabarti, C., “A new Viterbi decoder design for code rate k/n,” International Conference on Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 Volume: 4, http://dx.doi.org/10.1109/ICASSP.1995.480129

[9] Arunlal K. S., Dr. Hariprasad S. A., “An efficient Viterbi Decoder,” International Journal of Advanced Information Technology (IJAIT) Vol. 2, No.1, February 2012, Available at http://airccse.org/journal/ijcsea/papers/2112ijcsea09.pdf

[10] Chakraborty D., Raha P., Bhattacharya A., Dutta R., “Speed optimization of a FPGA based modified viterbi decoder,” International Conference on Computer Communication and Informatics (ICCCI), 2013, http://dx.doi.org/ 10.1109/ICCCI.2013.6466245

[11] Fei Sun, Tong Zhang, “Low-Power State-Parallel Relaxed Adaptive Viterbi Decoder,” IEEE Transactions on Circuits and Systems I: Regular Papers, Volume: 54 , Issue: 5, http://dx.doi.org/ 10.1109/TCSI.2007.890617

[12] Kuo C. J., Chang-Shyan Lin, “Viterbi-based algorithm for side-match vector quantization over noisy channels,” IEEE Transactions on Communications, Volume: 44 , Issue: 11, http://dx.doi.org/ 10.1109/26.544462

Copyright Disclaimer