Design of Low Power Asynchronous Viterbi Decoder for Wireless Applications

Mr. Girish D. Korde[1], Dr. Sanjay L. Haridas[2]
[1]Research Scholar, [2]Professor,
[1]B. D. College of Engineering, Sevagram, Wardha (M.S.), India
[2]G. H. Raisoni College of Engineering, Nagpur (M.S.), India

Abstract: In this paper design of low power asynchronous Viterbi decoder for wireless applications is exposed. Algorithm based designed which uses bundle data (BD) protocol is implemented here for code rate 1/2 and constraint length 3. Four phase bundled data handshake protocol is used as a communication protocol to synchronize the data between different blocks of Viterbi decoder. From the obtained results it is observed that hybrid register exchange method is being outperforms than register exchange method as this helps to reduce the switching activity and hence the dynamic power. Proposed asynchronous Viterbi decoder using HRE method helps to reduce average dynamic power by 61.65% than its synchronous counterpart. Complete design is implemented in Xilinx 13.1 and its XPower analyzer tool is used to calculate the dynamic power.

Keywords- Asynchronous Viterbi Decoder, four-phase, two-phase, Hybrid register exchange

I Introduction

Information theory plays a vital role in digital communication to achieve high efficiency and accuracy of the transmitting data. Convolution coding is being normally used in wireless communication for transmitting data over noisy channel. This noisy information can be decoded with the help of suitable decoder.

Viterbi [1] [2] decoder was firstly introduced in 1967 by Andrew J. Viterbi which was modified in 1971[3] by Forney considering it is the best suited for decoding the maximum likelihood sequence. Viterbi decoder is used to decode the convolution code in many applications which consumes large amount of power to decode the sequence.

With the updating technology power consumption is important parameter to be considered while designing the Viterbi decoder. Looking towards the demand of consumers request for low power high speed devices VLSI engineer is facing a big challenge. Reduced power consumption of portable devices will helps to improve the battery life of such devices. So to reduce the dynamic power of any system voltage scaling can be used but it faces the technology related problems. Also if frequency is increased, power consumption will increase which is not realistic solution. Reducing the switching activity of the system one can solve the problem related to dynamic power consumption. Further reduction of power consumption is possible with the aid of asynchronous design templates.

Most of the circuits are fabricated today are the synchronous which are governed by global clock this results in high power consumption of circuits. Asynchronous circuits are fundamentally different, these circuits use local handshake signal in order to perform necessary synchronizations. Though asynchronous designs [4] are not still well established and widely used design methodology. Asynchronous circuits inherent different properties that can be exploited to advantage in area, power, speed, EMI etc. As asynchronous circuits reduce the power consumption of the device, so VLSI designer have option to choose the design methodology for low power consumption of Viterbi decoder.

This paper is divided into several sections as; Section 2 gives a brief description about convolution encoder. Design of Viterbi decoder is given in section 3. Section 4 describes different decoding methods for convolution encoder followed asynchronous Viterbi Decoder in section 5. Results & discussion is appeared in section 6 and a conclusion will found in section 7.

II Convolution Encoder

The convolution encoder is basically a finite state machine used to encode the input sequence. Convolution encoder is used here with code rate 1/2and constraint length 3. Hence two memory elements will require designing encoder. This encoder will generate two bit output at every input bit. It consists of 2^{K-1} states in VD. Therefore, there are 2^{K-1} surviving paths at each stage, and 2^{K-1} metrics, one for each surviving path. The generator polynomial for the encoder is given by: $i_1 = [1, 0, 1]$ and $i_0 = [1, 1, 1]$. Generator polynomial specifies the connections of the encoder to the modulo-2 adder. The 1 in the generator polynomial indicates the connections and 0 indicates no connections between the stage and the modulo-2 adder as shown in figure. Encoder design will work in serially time shifted data sequence manner. It involves the modulo-2 addition to generate output to follow equation 1 and equation 2.

$$i_0 = inp \oplus FF_1 \oplus FF_0 \qquad ---\ equation\ 2.1$$

$$i_1 = inp \oplus FF_0 \qquad ---\ equation\ 2.2$$

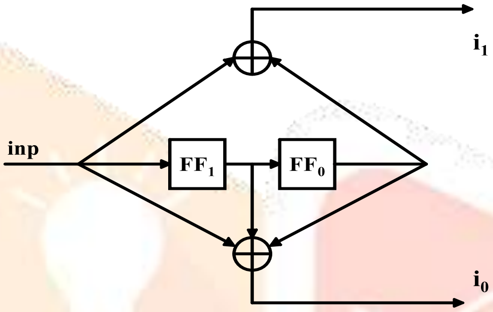

Figure 1: Convolution Encoder for (2, 1, 3)

From figure 1 it is cleared that convolution encoder for (2, 1, 3) is having two memory elements FF_1 and FF_0. As two memory elements are required to design the convolution encoder it consists of $2^{(K-1)}$ ie four states. Let us consider an input given to encoder is 110101011000. After performing modulo-2 addition it will generate the output sequence as 11 10 10 00 01 00 01 00 10 10 11 00. Convolution code can be graphically represented by three distinct methods viz: state diagram, tree diagram and trellis diagram. State diagram representation is shown in figure 2.

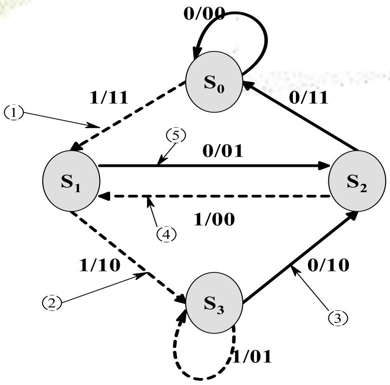

Figure 2: State Diagram Representation

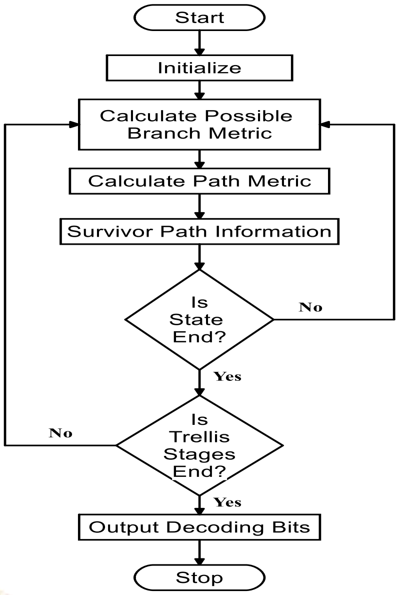

Figure 3: Design Flow of Viterbi Decoder

III Viterbi Decoder

Block diagram representation of asynchronous Viterbi decoder is depicted in figure 4. It consists of branch metric unit, add compare select unit and survivor memory unit. Each block in the design is controlled by request and acknowledge signal.

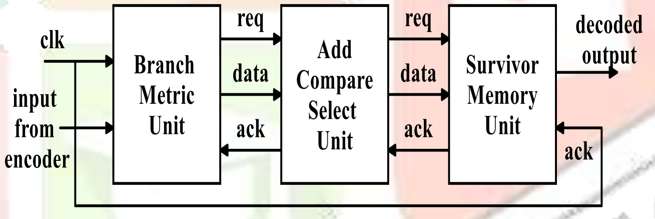

Figure 4: Block Diagram of Asynchronous Viterbi Decoder

A Branch Metric Unit (BMU)

Branch metric unit is the first block of Viterbi decoder here branch metric is calculated using hamming distance or Euclidian distance property depending on the decoding decision. Proposed Viterbi decoder designed is for soft decision. To calculate the branch metric; input sequence is fed directly to BMU block from encoder. Four bit branch metric is generated from two 3-bit soft decision input bits (i_0, i_1). Input bits are represented in two's complement representation. Simple add and subtract operation is performed by BMU on the input bits to generate output [5]. Adder calculates the Euclidean distance from received input codeword. For example, the branch metric for the state transition which produces the binary output (01) is (i_0 - i_1). Branch metric calculation using soft decision decoding is given in figure 5.

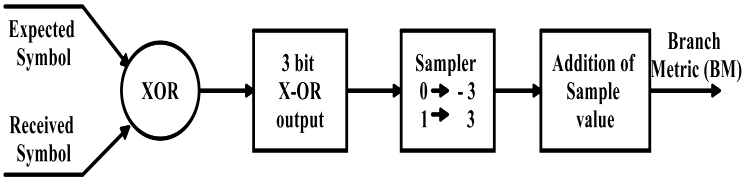

Figure 5: Branch Metric Calculation using Soft Decision Decoding

B Add Compare Select (ACS) Unit

ACS unit plays vital role in Viterbi decoder while calculating the new path metric. ACS Unit consists of adder, comparator and selector as depicted in figure 6. The ACS unit adds the BMs to the corresponding Path Metrics (PM). At each state there are two incoming paths. Both the path is having their corresponding path metrics. These two path metrics are compared by comparator and whichever is having high path metric is discarded by selector. And the path metric with minimum value is considered for the next computation is stored in PMM. ACS unit store associated survivor path decision in the SMU.

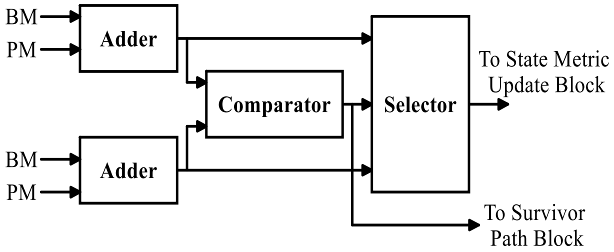

Figure 6: Block Diagram of ACS

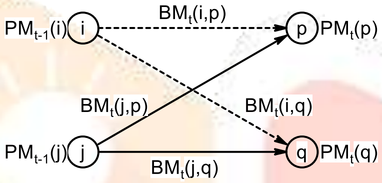

Figure 7: Butterfly Structure

$$PM_t(p) = \min\left(PM_{t-1}(i) + BM_t(i,p), PM_{t-1}(j) + BM_t(j,p)\right) \qquad \text{equation 3.1}$$

$$PM_t(q) = \min\left(PM_{t-1}(j) + BM_t(j,q), PM_{t-1}(i) + BM_t(i,q)\right) \qquad \text{equation 3.2}$$

ACS module in the Viterbi decoder needs to calculate the above equations continuously to get maximum likelihood path.

C Survivor Memory Unit (SMU)

The Survivor Memory Unit is a very important element of a Viterbi decoder design. Traditional SMU implementations use the register exchange or the trace back approaches. In Viterbi decoder, the Survivor Memory Unit is the final block. It keeps a record of the computations made on the input data, stretching back over many time slots from the current slot. This record enables the SMU to determine and output the most likely data based on the probable states of the encoder [6].

IV Decoding Methods

A Register Exchange (RE) Method

Register Exchange method [11] is simpler and commonly used technique. In this method SMU is constructed using bank of registers connected in same manner as the trellis diagram. As shown in Figure 8, register is assigned to each state contains information bits for the survivor path through the trellis. Register keeps record of partially decoded output sequence along the path. The register exchange method eliminates the need to trace-back since the register of final state contains the decoded output.

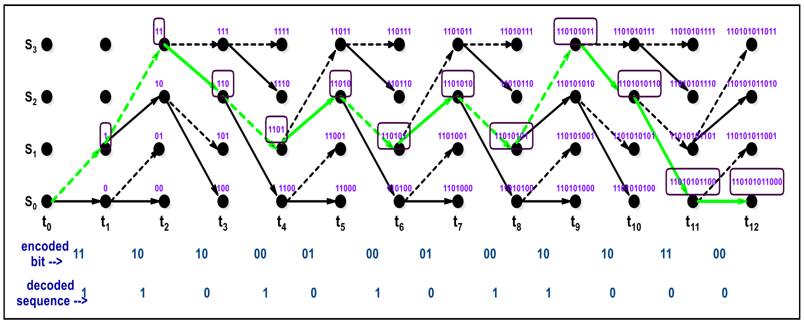

Figure 8: Example of Register Exchange Method

For instance, at time t_1 the survivor branch for S_1 state is from S_0 state at t_0; therefore, content of the S_1 state becomes "1" at t_1 and the corresponding decoded data for the survivor branch, which is a "1", is appended to it. At time t_2 the survivor branch for S_3 state is from S_1 state at t_1; therefore the initial content of S_1 state register, which is "1", is shifted into S_3 state register at t_2 and the corresponding decoded data for the survivor branch, which is "1", is appended to it. So the content of S_3 state register becomes 11. At time t_3 the survivor branch for S_2 state is from S_3 state at t_2; therefore the initial content of S_3 state register, which is "11", is shifted into S_2 state register at t_3 and the corresponding decoded data for the survivor branch, which is "0", is appended to it. So the content of S_2 state register at t_3 becomes 110. This process goes continue till the end of encoded bit, the final state register of S_0 contains the decoded output.

B Hybrid Register Exchange (HRE) Method

Drawback of register exchange method can be overcome by using HRE [11] method. Hybrid register exchange method is the combination of register exchange method and trace-back method, which reduce the switching activity and power. Property of trellis is being used to decode the sequence by using HRE method. By the property of trellis if we go forward for m cycles then the data bits will be the corresponding state address bits irrespective of the initial state from where the data gets transferred. To find the initial state one has to traceback through an m cycle by observing the survivor memory. And then transfer the partial decoded data from initial state to the next state which is m cycle latter and not a subsequent cycle. Memory operation is not in every clock cycle, and it gets reduced by factor m. In this way shifting of data from one register to another is reduced and hence the switching activity and ultimately power consumption will get reduced.

Figure 9 illustrate example of hybrid register exchange method. If the current state is S_3, the survivor state is $S_{2,}$ and the data input is 11. So the previous data from S_2 will get transferred to S_3 and new data is added as 11. From the figure 5 survivor state at t_4 is "11" for state S_1 therefore the content of S_3 state register at t_2 is shifted into S_1 state register at t_4 and the corresponding decoded data for the survivor branch, which is "01", is appended to it. So the content of S_1 state register at t_4 becomes 1101.

Survivor state at t_6 is "1101" for state S_1 therefore the content of S_1 state register at t_4 is shifted into S_1 state register at t_6 and the corresponding decoded data for the survivor branch, which is "01", is appended to it. So the content of S_1 state register at t_6 becomes 110101. This process goes continue, the final state register of S_0 contains the decoded output.

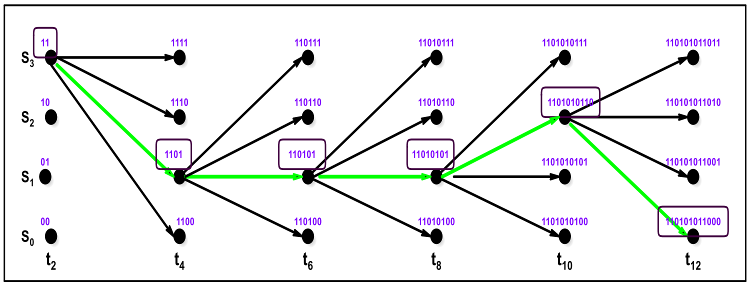

Figure 9: Example of Hybrid Registers Exchange Method

V Asynchronous Viterbi Decoder

Asynchronous logic has numerous important advantages over clocked synchronous logic. Asynchronous circuits works on average-case delay rather than worst-case delay and hence these are faster and dissipates low power because of no global clock. Asynchronous circuits have ability for low power and can be used to systems with low power dissipation.

Digital circuit is said to be asynchronous when no global or regionally global clock signal control any sequencing of events. Asynchronous circuits uses handshaking between its components to synchronize, communicate & operate. In handshaking two signals are used in opposite direction of each other called request and acknowledge for communication. Depending upon the application different handshaking protocol such as two-phase or four-phase handshaking can be used.

A Two Phase Handshake [12]

An asynchronous design supports the handshaking protocol. Two phase handshaking acts on rising or falling edge of local request signal. In this signaling scheme, the operation occurs is (1) data available (2) change request to high (active) state (3) change acknowledge to high (active) state. As shown in figure 1; when valid data is available request become 1, after delay acknowledgement become 1 and data can be transmitted. Two phase protocol is also called as non return to zero protocol as it is edge triggered. Therefore both the rising and falling edges of handshake signals show a new activity on the related signal. Two phase style can only be used in bundled data communication. Two phase systems can be faster than four phase as they uses non return to zero phases. But circuit require to implement this protocol is more complicated.

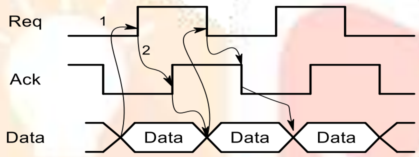

Figure 10: Two Phase Handshake Protocol

B Four Phase Handshake [12]

Four phase protocol [8] [13] is one of the methods of communicating between computing units in asynchronous system. Four phase bundled data protocol is most commonly used handshaking protocol. Activity while transmission of data using four phase handshake protocol is shown in figure 11. In this signaling scheme, the operation occur is (1) data available (2) change request to high (active) state (3) Change acknowledge to high (active) state (4) return request to low (inactive) state (5) return acknowledge to low (inactive) state [7].

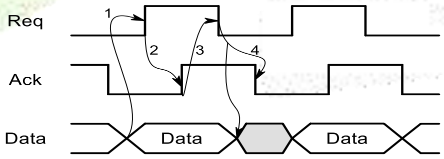

Figure 11: Four-Phase Handshake Protocol

When four-phase signaling is used there is a choice to be made as to which edge (rising or falling) of each handshake signal is active and takes the place of the event described above; the other edge is inactive and is part of the recovery phase during which the circuit prepares for the next cycle.

Implementation of asynchronous Viterbi decoder using hybrid register exchange method is represented in figure 12.

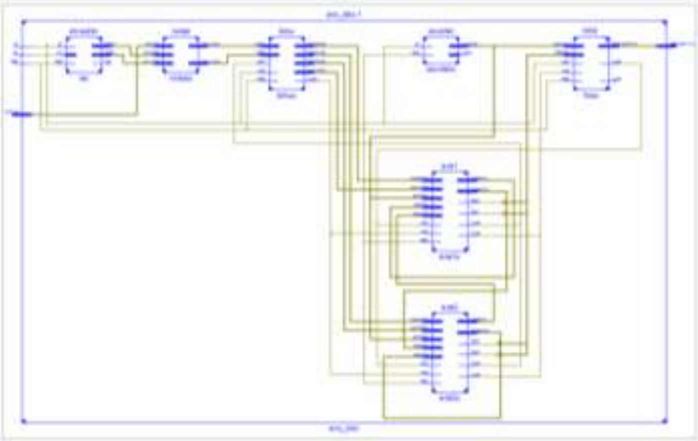

Figure 12: Asynchronous VD Using Four Phase Protocol for HRE Method

VI Result

The Viterbi decoder is designed for synchronous and asynchronous Viterbi decoder. The proposed design of Viterbi decoder is designed, simulated and synthesis using Xilinx 13.1 for the code rate 1/2 with constraint length 3. Figure 13 and figure 14 represents the output waveform of the asynchronous Viterbi decoder using four phase handshake protocol for RE and HRE method respectively.

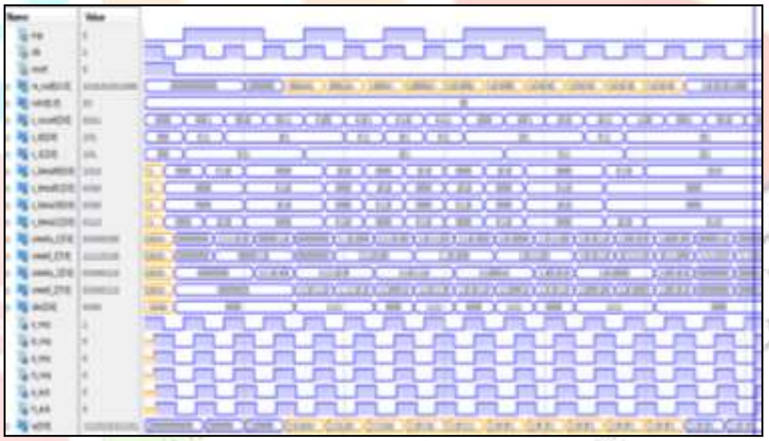

Figure 13: Simulation Waveform Using Four Phase Protocol for RE Method

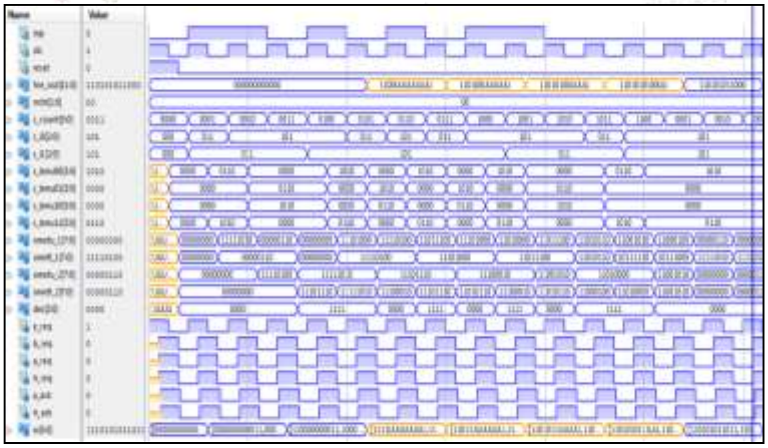

Figure 14: Simulation Waveform Using Four Phase Protocol for HRE Method

The power dissipation is the summation of the static and the dynamic power dissipation. The Design compiler refers to the characterization of the technology library to estimate the static power dissipation. It computes the dynamic power dissipation using the formula $P = \alpha*C_L*V^2 *f$, where α is the switching activity, C_L is the parasitic capacitance, V is the supply voltage, and f is the clock frequency. The capacitance and the supply voltage depend on the processing technology and are defined in the technology library.

Table 1: Comparative Analysis of Dynamic Power Dissipation (mW)

	Sync.	Async. GALS	Async. Pipeline	Async. 4-phase
RE Method (mW)	0.40667	0.306	0.27933	0.14733
HRE Method (mW)	0.346	0.304	0.222	0.13267

Comparative analysis for dynamic power dissipation of different architectures used to implement Viterbi decoder design is given in table 1. Here Viterbi decoder is implemented using register exchange and hybrid register exchange method to calculate the dynamic power dissipation. Asynchronous Viterbi decoder implemented using four-phase bundled data handshake protocol gives the lowest dynamic power dissipation as compare to other all architecture. Further power reduction of the decoder is achieved by reducing the switching activity in hybrid register exchange method. This approach reduces the dynamic power by 9.95% when compared to RE method in four-phase bundled data handshake protocol.

Table 2: Comparison of Asynchronous Viterbi decoder with the state of art from existing designs

	Ref 9 @ 145.12 MHz	Ref 10 @ 181 Mbps	our 4-Phase RE @ 200 MHz	our 4-Phase HRE @ 200 MHz
Dynamic Power (mW)	112.11	6.52	6.06	5.96

Table 2 gives the comparison between the asynchronous Viterbi decoder with the state of art from existing designs and our implemented design using RE and HRE methods. From the table it is observed that proposed implemented asynchronous Viterbi decoder using four-phase bundled data outperforms the previous designs with 8.58% decrease in the dynamic power.

VII Conclusion

In this paper we have designed low power asynchronous Viterbi decoder. This paper has presented improved performance of the Viterbi decoder for constraint length K=3 and code rate 1/2. The proposed asynchronous design reduces the dynamic power by 63.77 % than synchronous RE method and 61.65% over synchronous HRE method. The complete design is carried out using VHDL and power is calculated using XPower analyzer in Xilinx ISE design Suite 13.1.

References

[1] Andrew J. Viterbi, “Error Bound For Convolutional Codes and an Asymptotically Optimum Decoding Algorithm”, IEEE Trans. on Information Theory, Vol. IT-13, No, 2, April 1967, pg. 260-269.

[2] Andrew J. Viterbi, “Convolutional Codes and Their Performance in Communication Systems”, IEEE Trans. On Communication Technology, Vol. 19, Issue 5, October 1971, pg. 751-772.

[3] G. David Forney, “ The Viterbi Algorithm”, Proceedings of the IEEE, Vol. 61, No. 3, March 1973, pg. 268-278.

[4] Jens Sparsø, “Asynchronous Circuit Design”, A Tutorial, Technical University of Denmark, March 17, 2006.

[5] Dalia Abdel-Wahed Fouad El-Dib, “Low Power Register Exchange Viterbi Decoder for Wireless Applications”, PhD thesis Waterloo, Ontario, Canada, 2004.

[6] Wei Shao, Linda Brackenbury, “No- Handshake Asynchronous Survivor Memory Unit for a Viterbi Decoder”, 1-4244-1378-8/07, IEEE, 2007, pg 729-733.

[7] Sun-Yen Tan, Wen-Tzeng Huang, “A VHDL-based Design Methodology for Asynchronous Circuits”, WSEAS Transaction on Circuits and Systems, ISSN: 1109-2734, Issue5, Vol 9, May 2010, pg 315-324.

[8] C. Brej, J. D. Garside, “A Quasi-Delay-Insensitive Method to Overcome Transistor Variation”, Proceedings of the 18th International Conference on VLSI Design, VLSID’05, 1063-9667/05, 2005, IEEE.

[9] M. Santhi, Dr. G. Laxminarayanan and Surya Varadhan, “FPGA based Asynchronous Pipelined Viterbi Decoder using Two Phase Bundled-Data Protocol”, International SOC Design Conference, 2008, IEEE, pg I-314 to I-317.

[10] C. Arun, V. Rajamani, “A Low Power and High Speed Viterbi Decoder Based on Deep Pipelined, Clock Blocking and Hazards Filtering”, Int. Journal Communication, Network and System Sciences, 2009, 6, 575-582, DOI: 10.4236/ijcns.2009.26064.

[11] S. L. Haridas, Dr. N. K. Choudhari, “Design of Viterbi Decoder with Modified Trace-back and Hybrid Register Exchange Processing”, Int. conf. ACAC3’09, January 2009, ACM 978-1-60558-351-8.

[12] Matheus Trevisan Moreira, Ney Laert Vilar Calazans, “Proposal of an Exploration of Asynchronous Circuits Templates and Their Applications”, Technical Report Series,

[13] M. Kawokgy, C. Andre T. Salama, “Low Power Asynchronous Viterbi Decoder for Wireless Applications”, ISLPED-04, August 9-11, 2004, USA. ACM 1-58113-929-2/04/0008.

[14] Mohammad K. Akbari, Ali Jahanian, Mohsen Naderi, Bahman Javedi, “Area Efficient, Low Power and Robust Design for Add-Compare-Select Units”, proceeding of EUROMICRO Systems on Digital System Design (DSD’04), IEEE,0-7695-22033/04, 2004.

[15] Linda E. M. Brackenbury, “An Asynchronous Viterbi Decoder”, EU PREST project EP25242, AMULET Group, Department of Computer Science, University of Manchester.