

# Wave Pipelined VLSI Architecture for a Viterbi Decoder Using Self Reset Logic with 0.65nm Technology

Kalavathi Devi T<sup>a,\*</sup> and C Venkatesh<sup>b</sup>

<sup>a</sup> Senior Lecturer, Department of EEE, Kongu Engineering College, Perundurai, Erode., TN, India

<sup>b</sup> Dean, Faculty of Engineering, Erode Builders Institute of Technology, Erode., TN, India

**Abstract:** In 3G mobile terminals the Viterbi Decoder consumes approximately one third of the power consumption of a base band mobile transceiver. Viterbi decoders employed in digital wireless communications are complex and dissipate large power. A low power Viterbi decoder is designed in circuit level using self reset logic and wave pipelining technique is implemented for high speed operation. The Viterbi decoder consists of four units like branch metric unit, add compare and select unit and the survivor path memory unit. All these units are designed using the self reset logic and wave pipelining, and simulated with its layout using MICROWIND TOOL in the 0.65nm technology, 1.8V Vdd and at a frequency of 10GHz. The simulation result shows that the power consumption is reduced by 70.55% and the speed of the circuit is increased by 45.83% compared to the Single Rail Domino Logic for constraint length of  $K=3$  to 7.

**Keywords:** wave pipelining; self reset logic; Viterbi decoder; micro wind; and layout.

## 1. Introduction

Convolution coding and Viterbi decoding are widely used in modern digital communication systems, such as data communication, satellite communication, and mobile communication, to achieve low-error rate data transmission. The Viterbi algorithm is known to be an efficient method for the realization of maximum likelihood decoding of Convolution codes. Viterbi decoder is the hardware implementation of Viterbi algorithm. The great advantage of Viterbi algorithm [14] is that for a constraint length  $K$ , code rate  $r=k/n$ , where  $k$  is the number of inputs given to the encoder and  $n$  is the number of outputs taken from the encoder. The number of operations performed in the decoding of  $L$  bits is  $L * 2^{n(k-1)}$  which is linear in  $L$ . Architecture for the Viterbi decoder is designed using Self Reset

Logic (SRL) to reduce power dissipation. Wave Pipelining technique is designed to increase the speed of the architecture. Self Reset Logic [9] provides a design solution where the clocking overhead is minimized and the area and power consumption are reduced.

Wave pipelining [7, 9] is a method for the fast arithmetic circuit implementation since it renders high throughput and reduces the area and power by removing intermediate registers. Such registers results in latency due to their setup and clock-to-output times and introduce delays for each stage. Viterbi algorithm and various implementation techniques of Viterbi decoders have been investigated for the past three decades. Several works [6, 9,12] have been proposed for the Viterbi decoder mainly concentrating on the Add Compare Select

\* Corresponding author: E-mail: [kalavathidevi@gmail.com](mailto:kalavathidevi@gmail.com)

Accepted for Publication: October 26, 2010

(ACS) or Survivor Memory Unit (SMU). The basic Viterbi algorithm [3] was applied in digital communication systems, speech and character recognition. It focused on the operations and the practical memory requirement to implement the Viterbi algorithm in real-time. Based on the data generated and decoded [10] from the zero Hamming distance path, unnecessary computations in the Viterbi decoder was avoided. Speed and power were not considered. Low-power bit-serial Viterbi decoder chip [1] for the code rate  $r = 1/3$  and the constraint length  $K = 9$  (256 states) was implemented. The add-compare-select (ACS) module was based on the bit-serial arithmetic and implemented with the pass transistor logic circuit. The Scarce state transition (SST) scheme [11] employed a simple pre decoder followed by a pre encoder to reduce the transitions of the Viterbi decoder. Combining the concept of wave pipelining and self reset logic, [8] provided an elegant solution at high speed data throughput with saving in power and area. The concept of wave pipelining was explained in [4] and the multiplier circuit was designed. In [13] Reduced Slack Precharge half buffer based asynchronous multiplier was designed that also incorporates pipelining techniques. Timing calculation for delays can be viewed from this work. The trellis diagram for a rate  $1/n$  convolution encoder consists of butterfly structures. This structure contains a pair of origin and destination states, and four interconnecting branches as shown in Figure 1

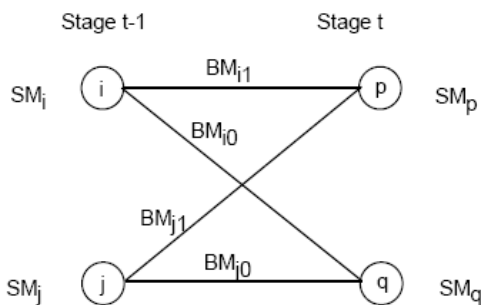


Figure 1. A Butterfly structure for a Convolution Encoder with rate  $1/n$

In Figure 1, the upper (lower) branch from each state  $i$  or  $j$  is taken, when the corresponding source input bit is '1' ('0'). If the source input bit is '1' ('0'), the next state for both  $i$  or  $j$  is state  $p$  ( $q$ ). The following relations in 1.1 are established for a  $(n,1,m)$  convolution encoder.

### 1.1. The relationships of the states and branch metrics in a butterfly structure:

For state I

$$j = 2^{m-1} + i \quad (1)$$

$$p = 2i + 1 \quad (2)$$

$$q = 2i \quad (3)$$

For branch metric  $BM_{xk}$

$$BM_{i0} = n - BM_{i1} \quad (4)$$

$$BM_{j0} = BM_{i1} \quad (5)$$

$$BM_{j1} = BM_{i0} \quad (6)$$

For state metric  $SM_x$

$$SM_p = \text{Min} [(SM_i + BM_{i1}), (SM_j + BM_{j1})] \quad (7)$$

$$SM_q = \text{Min} [(SM_i + BM_{i0}), (SM_j + BM_{j0})] \quad (8)$$

It is important to note that state  $p$  is even and state  $q$  is odd. This implies that an odd (even) state is reached only if the source input bit is '0' ('1'). Another important point to be noted is that, it is possible to trace back from a state at a stage  $t$  to its previous state at the stage  $t-1$  provided the survivor branch of the state is the upper path or the lower path. If the survivor branch of an odd state  $p$  at stage  $t$  is the upper (lower) path, the previous state at stage  $t-1$  is state  $i(j)$ . Note that  $i$  is obtained as  $(p-i)/2$  and  $j$  is  $2m-i+i = 2m-i + (p-i)/2$ . Similar results can be applied to an even state. In short, if it is recorded whether the survivor path is the upper path or the lower path, we can trace back from the final state to the initial state.

## 1.2. Problem Statement

A Viterbi decoder using Single Rail Domino Logic [8] was designed. Mainly the design is concentrated only on the Add compare select unit. For calculating the smallest path metric the mod module calculation is involved which increases the complexity of the hardware. Implementation of the circuits in dynamic domino logic has increased the gate capacitance that causes more switching. Hence in this proposed method self reset logic with wave pipelining technique is incorporated to ensure that unwanted switching is removed

## 1.3. Self Reset Logic

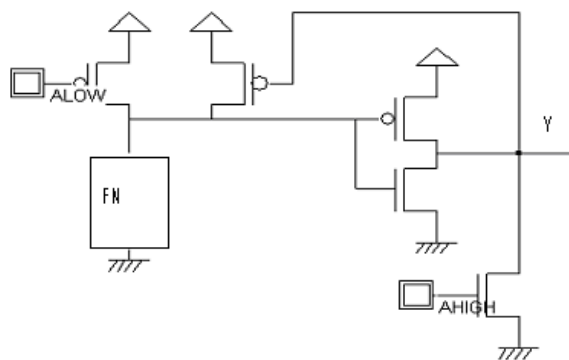


Figure 2. Basic circuit of Self reset logic

The above figure 2 shows the basic circuit of Self Reset Logic. It introduces the basic-concepts involved in the SRL cells. There is a sub block where the logic function performed by the gate is implemented. It is represented by the function [FN] block, which receives  $n$  input data pulses, where the circuit to be realized is implemented using the  $N$  transistors. The output of the gate  $Y$  provides a pulse if the logic function becomes TRUE. The reset signal is implemented as two separate pulses: RL (active low) and RH (active HIGH). RL is used to reset the input stage, while RH is used to reset the output stage after the output data has been propagated.

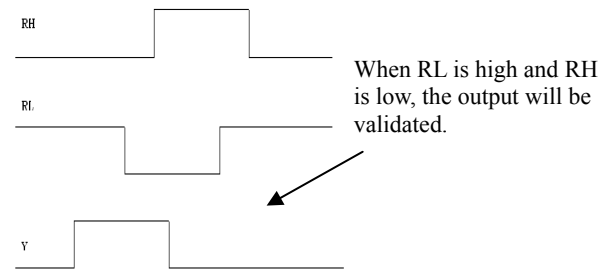


Figure 3. Timing diagram

Timing diagram is shown in figure 3, below. The basic operation of the SRL gate is as follows. The gate is initially in its standby state. In the standby state the power consumption is minimal. All the inputs are stable i.e., in the logic level low. When the inputs are received, some switching occurs and if the evaluated function becomes true, an output pulse is generated. No internal delay is generated by the circuit.

## 1.4. Basic Gates with SRL

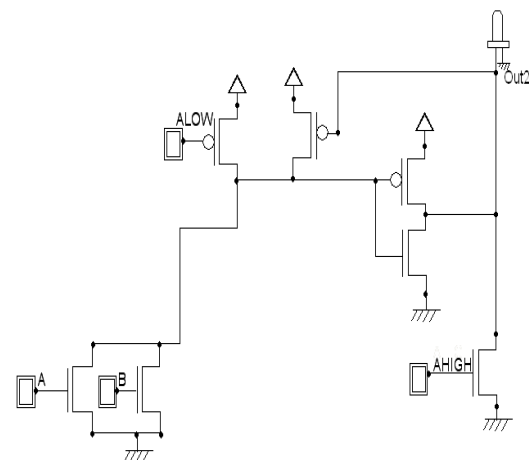


Figure 4. SRL based 2 input OR gates

The operation of the SRL based OR gate shown in Figure 4 is as follows, when  $A$  is high,  $B$  is low and  $ALOW$  is low, then the  $N$  transistor will turn on and the output high logic will be evaluated at the output. Similarly for other logics evaluation will take place in the same manner.

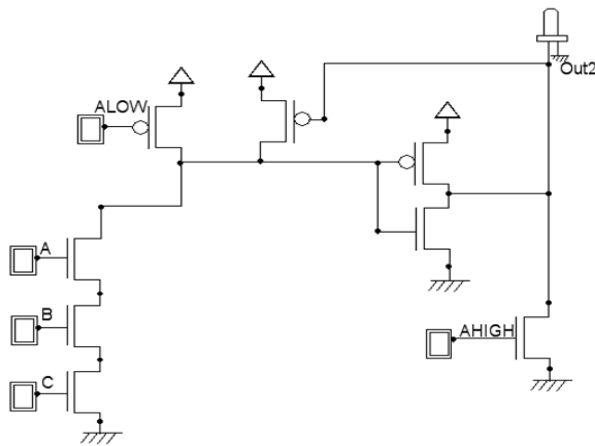


Figure 5 : SRL based 3 input AND gates

The operation of the SRL based AND gate is given in Figure 5: if A is high , B is low, C is low and ALOW is low, then the N transistor A will turn on , while the B and C transistors are in off condition. The output logic will be low. Similarly for other logics evaluation will take place.

## 2. Branch Metric Unit

In this method the convolution encoders are designed for different constraint length and the inputs to the Viterbi decoder are processed in the proposed SRL based architecture. The

branch metric computation block compares the received code symbol with the expected code symbol and counts the number of differing bits. If both the applied inputs are different either (high/low) or (low/high) then the output of the EXOR gate is HIGH. If both the input is same the output of the EXOR is LOW. If the AHIGH input is given to the NMOS transistor, it will reset the output and further changes in the input will not have any impact on the output of the circuit. If AHIGH='1' then OUT='0'. The 3-bit 1's counter as in figure.6 is designed using the asynchronous counter using the T-flip flop. The T-flip flop is the toggle flip flop and it consists of the reset and the clear input signal to make the flip flop asynchronous. Once the reset signal and the clear signal is given, the flip flop is in 000 counts. Now the clock input for the flip flop is given from the output of the EXOR gate. Whenever the EXOR gate output becomes HIGH the counter will be incremented by ONE, which computes the Branch Metric of the Viterbi Decoder. Now by applying the T input, T='1' the counter will start counting according to the clock. The T-flip flop is designed using D-flip flop and in turn the DFF is designed using the 3-input NAND gate.

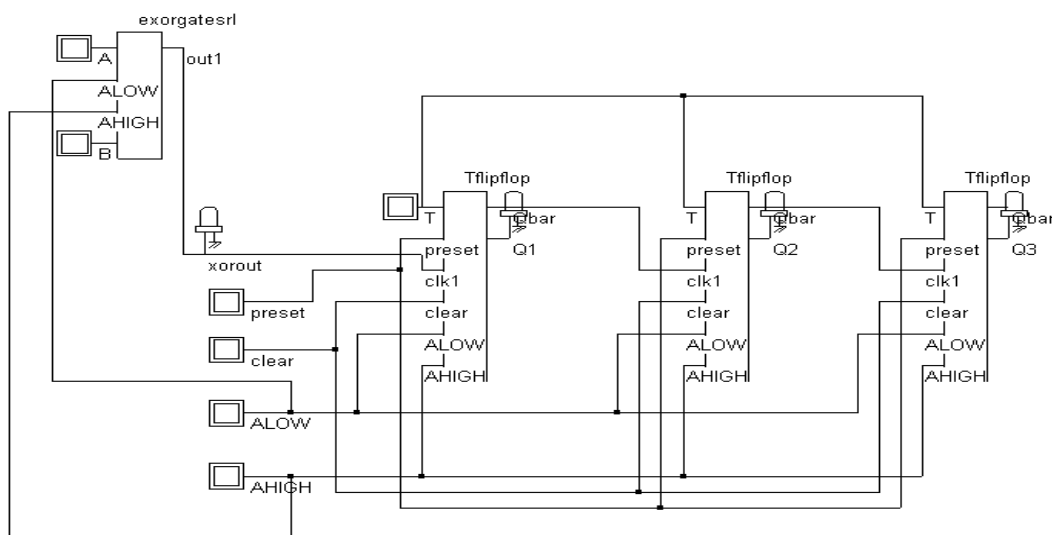


Figure 6. Branch Metric Unit Using SRL

The Branch Metric Unit is designed using the EXOR gate and the 3-bit counter. If the received code and the Expected code are different then the output of the BMU is high and it counts the number of 1's in the EXOR gate output. The output of the EXOR gate is fed as the clock input to the 3-bit counter. For example if the received code input and the expected code input differ by 5 bits, then the output of the BMU will be "101".

## 2.1. Add Compare and Select Unit

The adder unit which is proposed in the design consists of two full adders and one half adder. Output of the Branch metric unit (BMU) is added with the previous path metric and the obtained output is the new path metric for the next branch metric. Here  $P_0$  and  $A_0$  are the inputs to the half adder whose output is  $NP_0$  and  $C_0$ . This  $C_0$  is fed as an input to the next full adder along with  $A_2$  and  $P_01$ . The outputs of the adder unit are  $NP_0$ ,  $NP_1$ ,  $NP_2$  and  $NP_3$ . Thus, two SRL based Branch metric Units are designed. For the hardware implementation two adder levels are required. One adder for the upper path and the other adder for the lower path. Output of the adders is a four bit data and it is fed to the compara-

tor. Block diagram of the ACS unit is shown in figure 7. The outputs of the two adders are fed to the comparator unit, which compares the path metrics of all the branches. A Comparator unit is designed for three conditions. Each input is received by the comparator in a bit by bit manner. First output according to which that  $PM_1 < PM_2$  ( $A < B$ ) is considered and it is given to the selector unit. For Example if the two inputs to the comparator are  $A=1110$  and  $B=1111$  then the output line,  $A < B$  will go to the high state. Here the inputs  $A$  and  $B$  of the adder which has a lower value will be selected for the survivor memory unit ( $A < B$ ). If  $A=1111$  and  $B=1100$  then the output line  $A > B$  will go to the high state, which will be discarded. Each unit is designed by self reset logic which has two signals  $ALow$  and  $AHigh$ , indicated by arrows. The output of the comparator ( $A < B$ ) is given as the select signal for the multiplexer which is used to select the minimum path metric of the decoded message bit in the Viterbi decoder. The selector unit consist of four 2:1 MUX and the select signal for all the multiplexers are obtained from the  $A < B$  output of the comparator. Hence the selector selects the minimum path metric value. The Number of bits is represented by  $n$ .

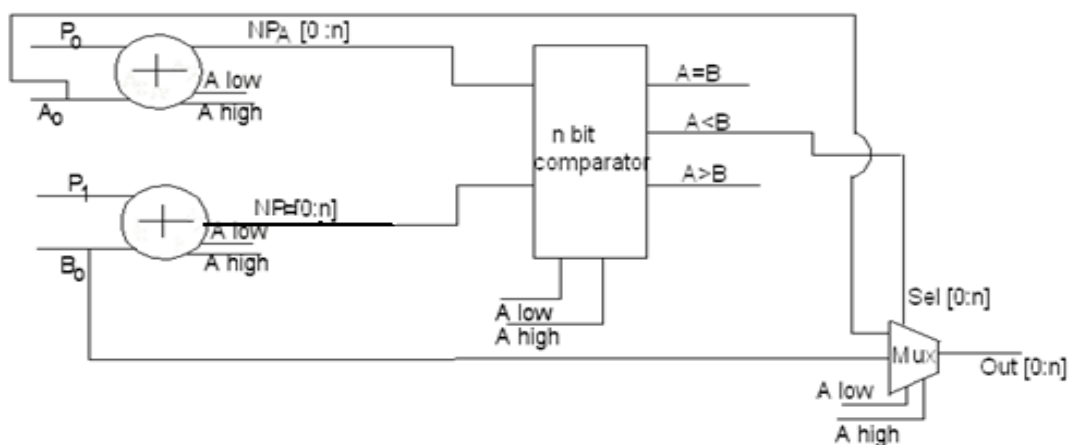


Figure 7. Proposed design of ACS unit with self reset logic

## 2.2 Survivor Memory Unit

In this paper, a modification of the RE method which avoids the power expense Register Exchange (RE) between two successive states is proposed. The RE method is based on successive RE operations between two origin states (i, j) and two destination states (p,q), using the butterfly unit. The Survivor memory unit can be designed by modified register exchange method [4] using the serial in serial out shift register and the length of the shift register depends on the length of the convolution encoder. For a constraint length of  $K=3$ , the survivor paths are determined by  $2^{K-1}$  where  $K$  is the constraint length. Figure 8 shows the circuit schematic for the Survivor memory unit using Self Reset Logic for a four bit combination. From the multiplexer in the comparator unit, the output goes to the shift registers. Initially the value is 0. Once the minimum path is traced the values of the signal  $A < B$  will acts as a pointer and

the concerned data will be stored in the registers. Four parallel stages of the SMU unit are designed for  $k=3$  as no more than  $2^{K-1} = 4$  stages of the survivor paths are required.

## 3. Wave Pipelined Viterbi Decoder With Self Reset Logic

All the blocks of the Viterbi decoder are designed using self reset logic. To implement the wave pipelining concept [9] all the inputs are fed through flipflops. Synchronization is done by adding buffers in the path. The timing requirements of wave-pipelined circuits as in figure 8 will be defined for a single combinational logic block with registers attached to its inputs and outputs. In the case of multi-stage pipelines, the timing constraints must hold good for all stages. Clock period can be reduced as long as data from a particular clock cycle does not overwrite data from the previous clock cycle.

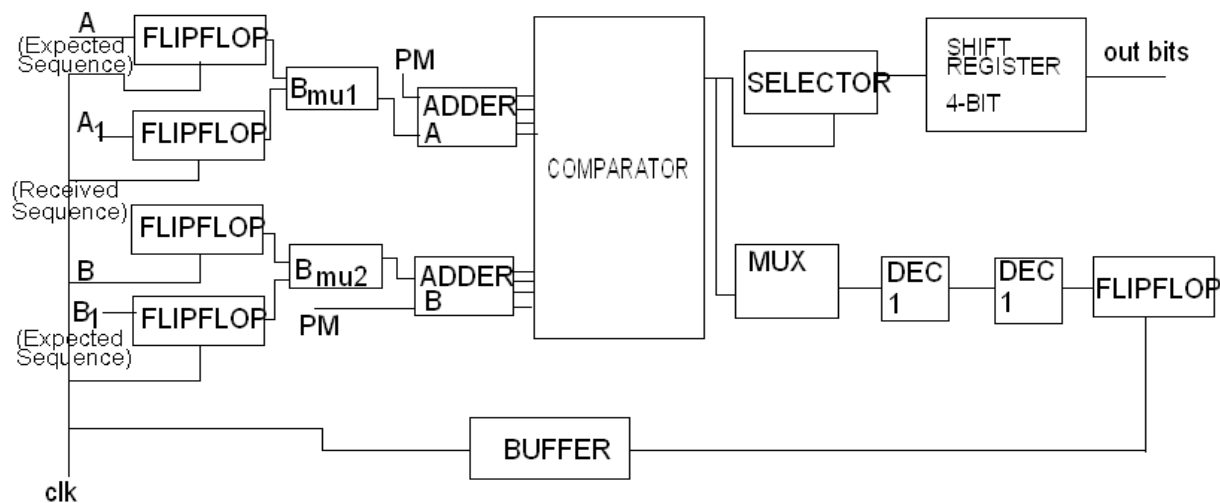


Figure 8. Design of the Wave Pipelined Viterbi Decoder with Self Reset Logic

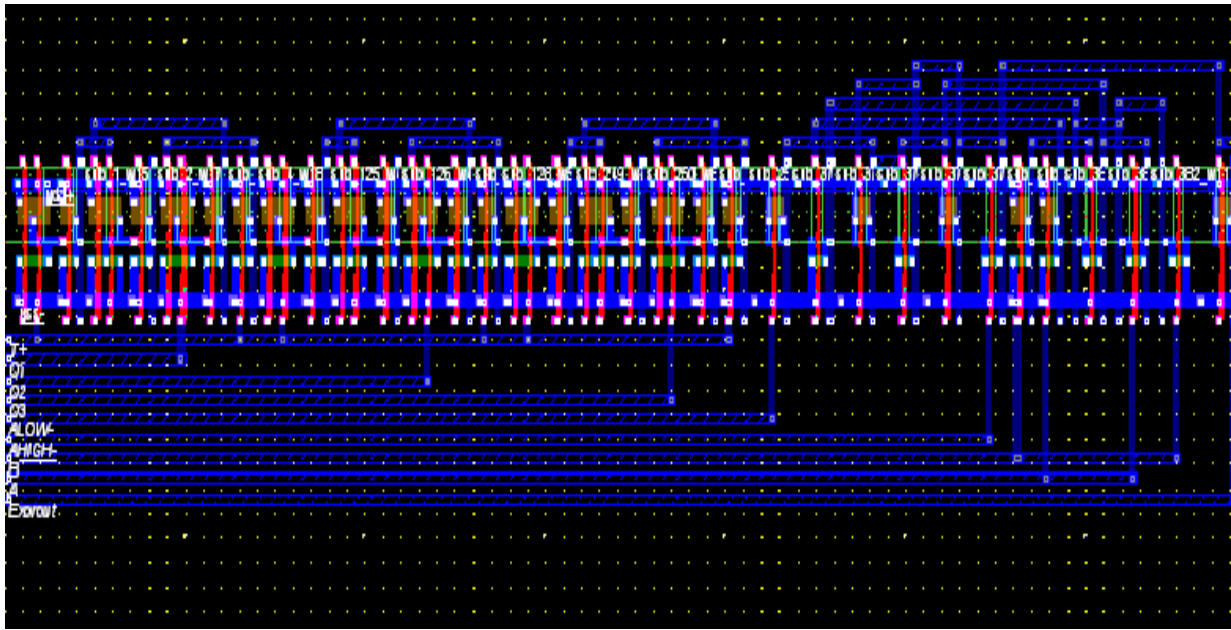


Figure 9. Layout design of the Wave Pipelined Viterbi Decoder with Self Reset Logic from Microwind tool

The designed circuit at the layout level is shown in figure.9. The internal design is given in figure.8 in the form of block diagram. Wave pipeline concept is implemented by placing registers at the input and output of the circuit. In order to balance the delay, the buffer is added at the critical paths of the design.

In this design all the Ahigh and Alow signals are set to low values and the input values are set by PWL window of the input signals. The no ideal effects like setup and hold times, clock skew at the input and output registers should also be considered. The clock for wave pipelined Tclk design can be expressed as follows as in equation 9

The clock period in a wave pipelined system is constrained by the difference between the maximum and minimum combinational logic delay. Further, at any internal node of the network, there must be minimum separation between the arrival time of the latest signal of a given wave, and the earliest signal of the next wave. All the values are obtained from the simulation result of the microwind tool. The frequency versus time window in the tool gives the delay values of the design. The maximum delay through the logic, including the clock overhead and clock skew is given by equation 10

$$T_{clk} > T_{delay}(\max) - T_{delay}(\min) + T_s + T_H + 2 * T_{skew} \quad (9)$$

$$T_{delay}(\max) = DR + D_{max} + T_s + \Delta_{ck} - \Delta \quad (10)$$

The minimum delay through the logic is expressed in equation 11 as,

$$T_{min} = DR + D_{max} - T_H - \Delta_{ck} - \Delta. \quad (11)$$

Calculation of timing delays is given as follows:  
From the results the calculations are given as: From equation 10. Delay maximum is

$$\begin{aligned}
T_{\text{delay (max)}} &= 0.02 + 0.03 + 0.001 \times 10^{-6} + 0.01 - 0.002 \\
T_{\text{delay (max)}} &= 0.058000001 \mu\text{s} \\
T_{\text{delay (min)}} &= 0.02 + 0.03 - 0.001 \times 10^{-6} - 0.01 - 0.002 = 0.037999999 \mu\text{s}.
\end{aligned}$$

$$\begin{aligned}
T_{\text{clk}} &> T_{\text{delay (max)}} - T_{\text{delay (min)}} + T_{\text{setup}} + T_{\text{hold}} + 2 \times T_{\text{skew}} \\
T_{\text{clk}} &> 0.058000001 - 0.037999999 + 0.001 \times 10^{-6} + 0.001 \times 10^{-6} + 2 \times 0.002 \\
T_{\text{clk}} &> 0.024000013 \mu\text{s} \\
T_{\text{clk}} &= 240 \times 10^{-2} \mu\text{s}
\end{aligned}$$

## 4. Simulation Results and Comparison

### 4.1. Output of Viterbi decoder without wave pipelining

Before implementing the wave pipelining technique, the results of the Viterbi decoder is shown in figure 10. This design is the implementation of Self Reset logic based Viterbi decoder. Convolution encoders are designed for the constraint lengths from K=3 to 7. The designed architecture is used for the analysis of outputs from various constraint length and results are shown below. The received code symbol is “0100110110110” and the expected code symbol is “0010011100011”. Normally the received code is the sequence obtained from the trellis diagram of the Viterbi algorithm. Expected code sequence is the user defined sequence. These inputs are fed to the EXOR gate (SRL) and the output shows the difference in the code symbol. This output is fed to the 3-bit counter (SRL) to count the number of ones. The output of the counter shows that there are seven numbers of ones. This implies that the difference between the received and expected code symbol is 7(111). The output of the BMU and the previous Path Metric are added and the PM for all the branches of the Decoding algorithm is compared by the comparator. The least branch is selected to be stored in the Survivor Memory Unit. The selected path value is stored in the Survivor Memory Unit which is the serial in serial out shift register. If for example the input for the SMU is “1101” with in the four clock cycle the values are stored in the regis-

ter. Without the pipelining design the speed of the circuit is minimum. For Example, if the input to the decoder is 11 00 11 then the Decoder decodes the received value and the output of the decoder will be the transmitted input. The layout output without pipelining shows the power consumption of the Viterbi decoder is 299.082mW for a constraint length of K=3.

### 4.2. Layout Output of Wave Pipelined Viterbi Decoder with SRL

In the result shown in Figure 11. A1 refers to the received sequence 0011100101 and A2 is the expected sequence 1110101101 for the Branch metric 1. Similarly B1 is 0011101011 and B2 is 1011011101 for the Branch metric 2. All the signals such as clear, preset, T values are set to high. Area of the design is 36mm<sup>2</sup> with an operating voltage of 1.8V. Inputs to the adders are from the branch metrics whose outputs are 100 and 101. On comparing the two branch metrics Branch metric 1 has a lower value. This sequence will be selected by the selector. Out1, 2, 3 and 4 of the SMU has the results which is 0011100101. In implementing the wave pipelining technique, this result is stored in the dec1 and dec2 registers which is synchronized by the clock pulses of the input. Timing details are measured from the Frequency vs. time window of the Microwind tool. For the clock control, Maximum delay is 0.058s; minimum delay is 0.037s and Rise delay is 0.001μs. The clock period required for the entire operation is obtained as 0.024μs.

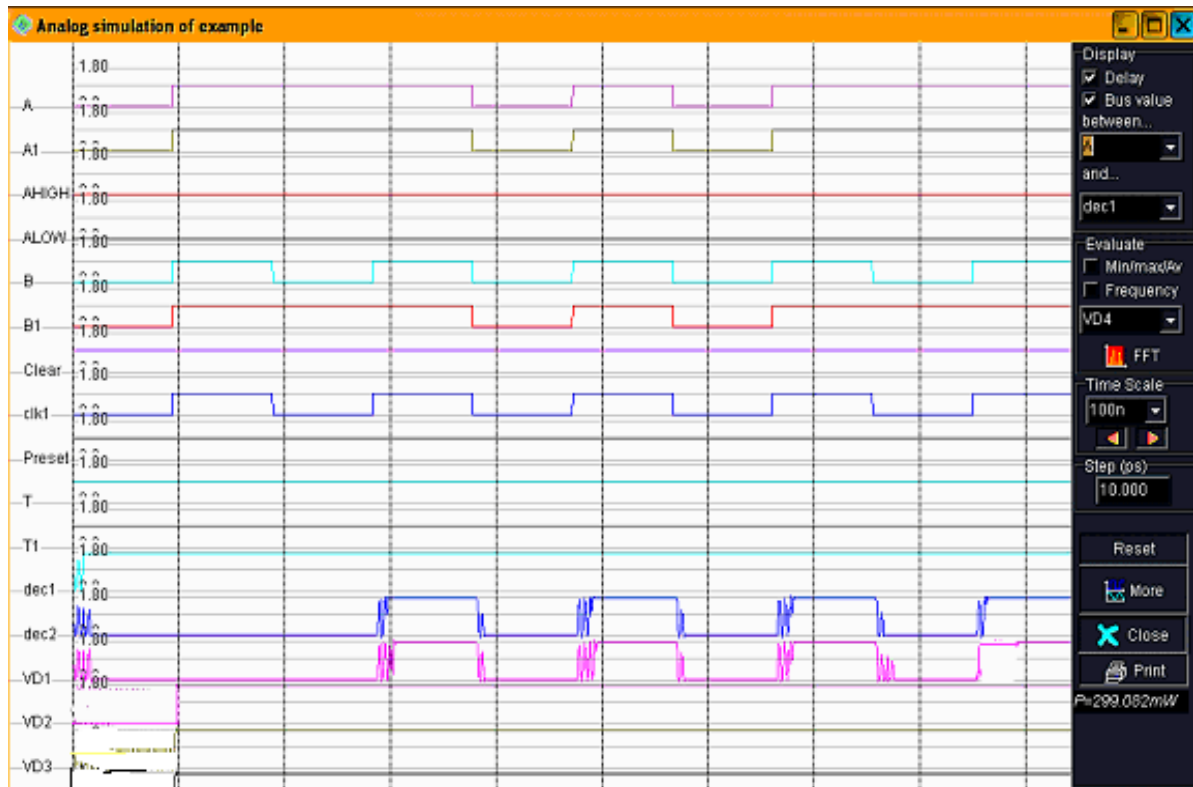


Figure 10. Output of Viterbi decoder without wave pipeline

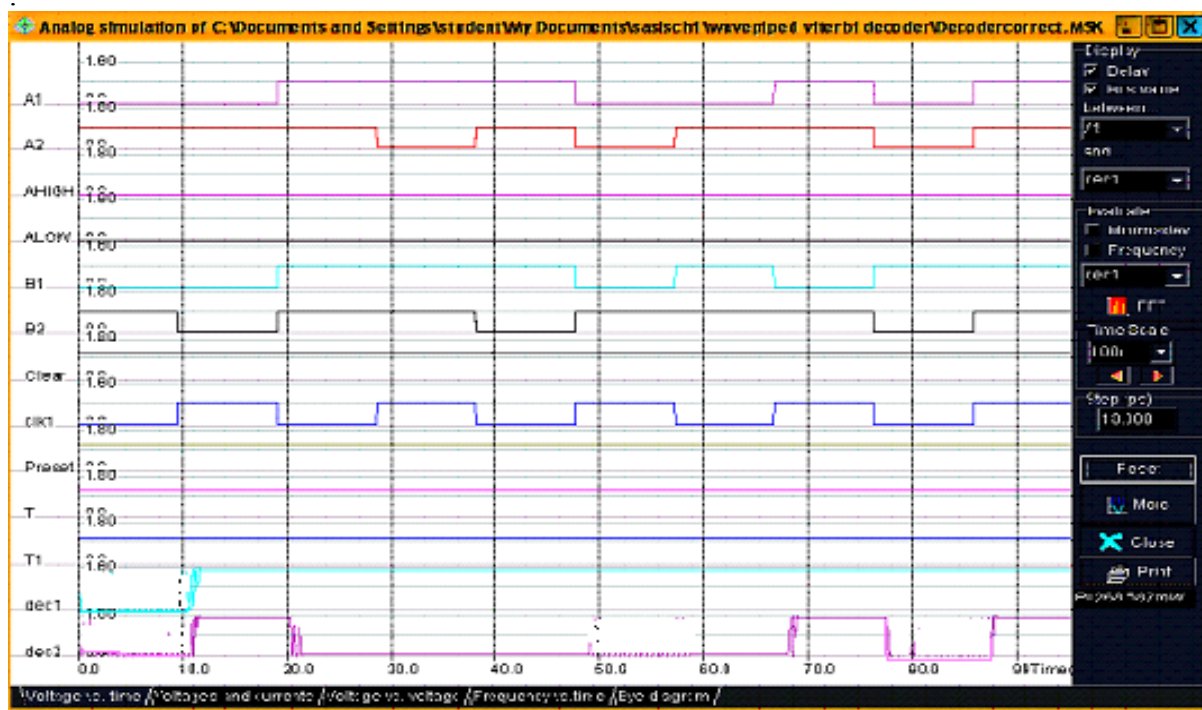


Figure 11. Layout Output of Wave pipelined Viterbi Decoder with SRL

Comparison of the proposed design with the single rail domino logic in terms of power is shown in Table 1. At 10 GHz frequency and 1.8V supply voltage the Viterbi decoder is simulated for various constraint lengths by designing the encoders and the results are

tabulated. Proposed design and existing method for comparison [9] are simulated at the layout level using MICROWIND tool with 0.65nm technology for various constraint length K=3 to 7 with a code rate of  $\frac{1}{2}$ .

**Table 1.** Performance Analysis of Power for a frequency of 10GHz for code rate  $\frac{1}{2}$

Input data (random generation)	Single rail domino logic		SRL with wave pipelining	
	Constraint length (K)	Total Power(w)	Constraint length (K)	Total Power (w)
0011100101	3	970mW	3	268.58mW
111 00 0110	4	982 mW	4	258.5mW
0100101000	5	923.5mW	5	259mW
1010100101	6	969mW	6	263mW
0100000100	7	983mW	7	258mW

Unwanted transitions are avoided in the self reset logic design. Inputs data's are randomly given for the design; here the number of bits taken is 10. For each constraint length a sample of 5 sets of 8 bit data is given as input to the decoder (which is the output of the encoder). In the table 1. for each constraint length a single sample data and its power consumption are given. The results show that the average Power consumption of the Viterbi decoder for a frequency of 10GHz is reduced by 72.31% compared to the single rail Domino Logic. It is hard to compare with all existing methods, because each method is targeted for different applications. In terms of power the proposed design outperforms the previous design with an increase in processing speed.

Table 2 shows the performance analysis of speed of the wave pipelined Viterbi decoder and Single rail domino logic. The result shows that the speed of the proposed method is increased.

The speed of the Wave pipelined Viterbi Decoder is increased by 45.83% compared to the non pipelined Viterbi Decoder for all the constraint lengths. Maximum operating

frequency of the wave pipelined Viterbi decoder is 10GHz.

**Table 2.** Performance Analysis of Speed

Module Name	Delay	
	Single Rail Domino Logic	SRL With Wave Pipe- lining
Viterbi Decoder	0.480ns	0.260ns

## 5. Conclusion

The Viterbi decoder is designed in circuit level using Self Reset Logic and is simulated with its layout using MICROWIND tool in the 0.65nm technology, 1.8V Vdd and at a frequency of 10GHz. The work focuses only on low power design. The Average power consumption of the Viterbi Decoder using Self Reset Logic is 299.082mW. Wave pipelining of the Viterbi Decoder is done and simulated with the Layout Tool. The result shows that the average power consumption of the wave pipelined viterbi decoder with SRL is 260.5mW. The simulated result is compared with the Viterbi decoder designed using

Single Rail Domino Logic. Simulation results proves that the average power consumed by the Viterbi Decoder is 70.55% less and Speed is increased by 45.83% compared to Non-pipelined Viterbi Decoder. Obtained results indicate that Self Reset Logic consumes less power compared to the other MOS Logic styles

## Appendix notations

Tdelay (max) = maximum delay of the clock

Tdelay (min) = minimum delay of the clock

Ts = setup time of clock

TH = Hold Time of clock

$\Delta$ ck = change in the clock period

$\Delta$  = phase shift ck and outDR – Rise delay

The parameters like DR, Dmax, Ts,  $\Delta$ ck are obtained from the frequency versus time

## Reference

- [ 1 ] Tsui, C. Y., Cheng, R. S., and Ling, K. 1999. Low power ACS unit design for the Viterbi decoder. *Proceedings of the IEEE symposium on Circuits and Systems*: 137-140.
- [ 2 ] Dalia, A. El-Dib and Elmasry, M. I. 2004. Modified Register-Exchange Viterbi Decoder for Low-Power Wireless Communications. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51, 2: 371- 378.
- [ 3 ] Forney, G. 1973. The Viterbi Algorithm. *Proceedings of the IEEE*.61, 3: 268-278.
- [ 4 ] Hauck, B. and Huss, B. 1998. Asynchronous wave pipelines for high throughput datapaths. *Proc. IEEE International. Conf. on Circuits Syst.*1: 283–286.
- [ 5 ] Lou, H. L. 1995. Implementing the Viterbi Algorithm. *IEEE Signal Processing Magazine*.
- [ 6 ] Kong, J. J. and Parhi, K. K. 2004. Low-Latency Architectures For High-Throughput Rate Viterbi Decoders. *IEEE Transactions on Very Large Scale Integration Systems*, 12, 6.
- [ 7 ] Litvin, M. E. and Mourad, S. 2005. Self-reset logic for fast arithmetic applications. *IEEE transactions on Very Large Scale Integration systems*. 13, 4: 462-475.
- [ 8 ] Litvin, M. and Mourad, S. 2006. Wave Pipelining with self reset Logic. *IEEE International Conference on Electronic Circuits and Systems. ICECS Nice, France*.
- [ 9 ] Niko, B. and Elisabeth S. 2004. A 2.8 Gb/s, 32-State, Radix-4 Viterbi Decoder Add-Compare-Select Unit. *International symposium on VLSI circuits digest of technical papers*. 170-173.
- [10] Shao, W. and Brackenbury, L. 2008. Pre-Processing of Convolutional Codes for reducing Decoding Power Consumption. *Proceedings of the IEEE*.
- [11] Jin, J. and Tsui, C. Y. 2007. Low-Power Limited-Search Parallel State Viterbi Decoder Implementation Based on Scarce State Transition. *IEEE Transactions on Very Large Scale Integration (VLSI) systems*. 15, 10: 1172-1177.
- [12] Goo, Y. J. and Lee, H. 2008. Two Bit-Level Pipelined Viterbi Decoder for High Performance UWB Applications. *IEEE International Symposium on Circuits and system*. 18-21.
- [13] Senthilkumar, A. and Natarajan, A. M. 2008. Design of high speed Asynchronous pipelined FIR Filter using Quasi delay Insensitive reduced Slack Pre-charged half buffer. *International journal of Applied Science and Engineering*.6, 2: 181-197.
- [14] Proakis, J. G. 2000. *Digital Communications*. McGraw Hill Higher Education. New Ed. Edition.
- [15] Shaker, S. W., Elramly, S. H., and Shehata, K. 2009. Design and implementation of Low Power Viterbi decoder for Software defined Wimax receiver. *Proceedings of 17<sup>th</sup> Telecommunications forum TELFOR, Belgrade*: 468-471.