International Journal of Electronic and Electrical Engineering.
ISSN 0974-2174, Volume 7, Number 5 (2014), pp. 483-490

http://www.irphouse.com

High Speed Convolution Encoding and Viterbi Decoding Using Dynamic Shift Register

Mohammad Javeed[1] and Battula Swapna[2]

[1]Orbit Technology Research Pvt .Ltd
[2]Pakabanda Bazar, Khammam, Andhra Pradesh.

Abstract

A convolution encoder and viterbi decoder is shown in this paper which will be useful for high speed applications. While performing the encoding and decoding of bits, shifting of bits either it may to left or to right be necessary. The timing and complexity increases while performing the shifting operation when the number of bits increases. To overcome this we are proposing a dynamic shift register for convolution encoding and viterbi decoding. The proposed shift register shifts four bits at a time. The implementation is for code rate ½, constraint length 9 and the implementation of viterbi algorithm is by using the hamming distance instead of Euclidean distance. By using the hamming distance the complexity of the system decreases. The proposed architecture decreases the power consumption by 72% approximately. We also discuss the timing analysis of the system. The code written in verilog and synthesized in Xilinx 13.2 version.

Keywords: Shift register, Convolution encoder, Viterbi decoder, hamming distance, Euclidean distance, multi bit flip flop, single bit flip flop.

1. Introduction

The use of Convolution encoder with probabilistic decoding can significantly improve the error performance of a communication system [1]. The Viterbi algorithm, which is widely used decoding algorithms, is optimal, but its complexity in both number of computations and memory requirement exponentially increases with the constraint length k of the code. Hence when the codes with a longer constraint length are required in order to achieve a low error probability, decoding algorithms whose complexity does not depend on k becomes attractive [2]. Several multiple paths, breadth first

decoding algorithms, such as M-algorithm [3], simmon's algorithm have been proposed to the alternatives of the Viterbi algorithm [4]. Unfortunately, with these algorithms, should the correct path be lost, then its recovery is rather difficult, leading to very long error events. The error propagation is usually contained by organizing the data in frames or blocks with a known starting state or by using some special recovery schemes [5].

In the proposed design we are proposing a shift register which shifts the 4 bits by using the multi bit flip flops. When the convolution encoding process is going on for constraint length 9, the shifting of bits is necessary. When shifting the bits one by it takes time for the encoding process. Here we are proposing a solution for the encoder which efficiently decreases the power consumption by 72%. The proposed design uses hamming distance instead of Euclidean distance to avoid the complexity.

General solutions for low power viterbi decoder (VD) design have been studied by existing work [6]. Power reduction in VD could be achieved by reducing the number of states or by over scaling he supply voltage [7]. Over scaling of the supply voltage usually needs to take into the whole consideration the whole system that includes the VD, at which we are not focusing at our research. In practical application RSSD (Reduced Sate Sequence Detection System) is more commonly used than M-algorithm which is generally not as efficient as M-algorithm [8] [9]. How ever searching for the optimal PM in the feedback loop still reduces the decoding speed.

This paper is scheduled as section II represents the Convolutional encoder. Section III represents the proposed system. Section IV shows about the viterbi decoder. Section V represents the simulation and Section VI concludes the paper.

2. Convolutional Encoder

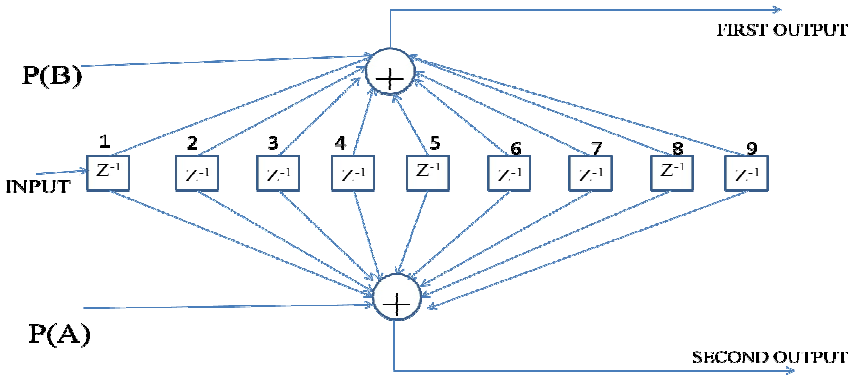

Fig. 1: Convolution encoder.

The Convolutional encoder with code rate ½ is shown in figure 1. Constraint length for the encoder is 9 as shown in the figure 1. The Convolutional encoder process is input is given to two xor gates. We choose the two polynomials with the same constraint length as P(A) and P(B). One xor gate performs it's operation with input and P(A) and second xor gate performs the operation between input and P(B).

Then we get the two outputs. Here in this process to shift the bits input as well as of the polynomials we require a shift register, which will be discussed in the next section.

3. Proposed System

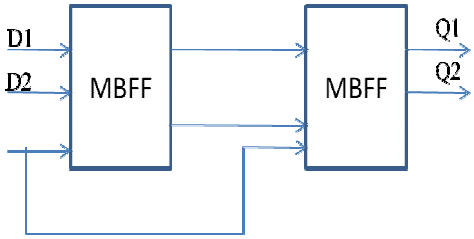

Fig. 2: MBFF shift register.

Multi bit flip flop:

Generally for the storage of bits the memory elements are latches and flip flops. The flip flop casually stores a single bit value. Here in our proposed system the flip flop stores the multi bits. Finally we designed a shift register by using the multi bit two flip flops which shifts the four bits. The diagram for the shift register is shown in the figure 2. The multi bit flip flop works under the technique of merging the clock pulses [10]. The timing diagram for multi bit flip flop is shown in figure 3 which is having the same operation as single bit flip flop. D-flip flop latches all the inputs to the output when the clock is high (for active high), or when clock is low (for active low). In an inactive state for both the cases the input holds the data. The proposed shift register is used for the shifting of bits according to the constraint length of the encoder for Convolutional encoder and viterbi decoder.

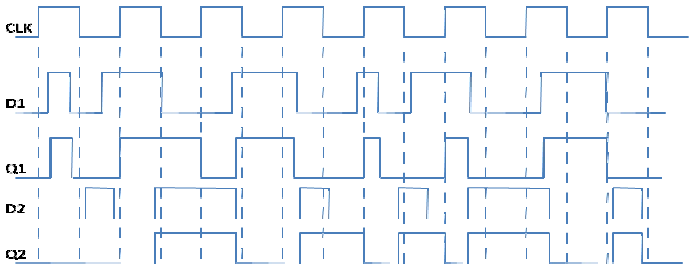

Fig. 3: Timing diagram for the multi bit flip flop.

4. Viterbi Decoder

The encoded bits by the Convolutional encoder are decoded by using the viterbi decoder. The block diagram for the viterbi decoder is shown in the Figure 4.

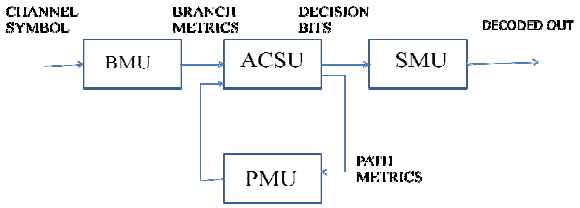

Fig. 4: Viterbi Decoder block diagram.

First branch metrics (BMs) are calculated in th BM unit (BMU) from the received bits.. Then BMs are fed into the ACSU that recursively computes PMs and output decision bits for each possible state transition. After that the decision bits are stored in and retrieved from the SMU in order to decode the source bits along the final survivor path. For decoding of Convolutional codes we can observe two types, soft decision decoding and hard decision decoding. We are concentrating on hard decision decoding.

Branch Metric Computation

The branch metric is a measure is a measure of the distance between what was transmitted and what was received and is defined each are in the trellis. In hard decision decoding, where we have given a sequence of parity bits, the branch metric is the hamming distance between the expected parity bits and the received bits. As example is shown in Fig. 5, Where received bits are 00. For each state transition, the number on the arc shows branch metric for its transition. Two of the branch metrics are 0, corresponding to only states and transitions where the coresponding hamming distance is 0.

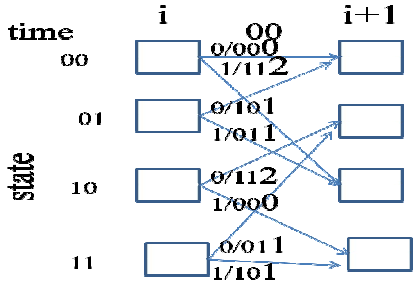

Fig. 5: Branch metric calculation.

Path metric calculation:

Suppose the receiver has computed the path metric PM[*s* ,*i*] for each state *s* (of which there are 2^{k-1} ,where k is the constraint length) at time step i, The value of PM[*s* , *i*] is the total number of bit errors detected when comparing the received parity bits to the most likely transmitted message, considering all messages that could have been sent by the transmitter until time step *i*(starting from state '00' , which we will take by convention to be the starting state always).

Among all possible states at time step *i*, the most likely state is the one with the smallest path metric. I f there are more than one state, they are all equally possibilities.

Now we determine the path metric at time step *i+1,* PM[*s,i+1]* ,First observe that if the transmitter is at state *s* at time step *i+1,* then it must have been in only one of the two possible states at time step *i.* These two predecessor states, labeled α and β, are always same for a given state. In fact, they depend only on the constraint length of the code and not on the parity functions. Fig. 5 shows the predecessor states for each state. For instance, for state 00, α=00 and β=01; for state 01, α=10, β=11.

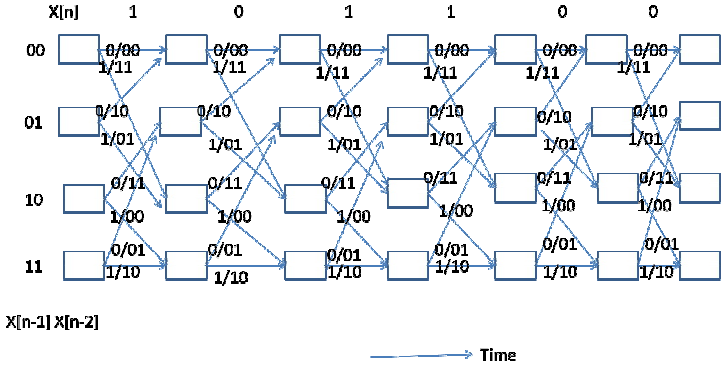

Fig. 6: Trellis diagram for the decoding.

Any message sequence that leaves the in state s at time *i+1* must have left the transmitter in state α or in a state β at time *i* . To arrive in state '01' at time *i+1,* one of the two properties must hold are: The transmitter was in state '10' at time '*i*' and the i^{th} message bit was 0. If that is the case, then the transmitter sent '11' as the parity bits and there were two bit errors, because e received two bits 00. Then the path metric of the new state, PM ['01', *i+1]* is equal to PM ['10', *i+2],* because the new sate is '01' and the corresponding path metric is larger by 2 containing the two errors.

The other possibility is that the transmitter was in state'11' at time *i* and i^{th} message bi was 0. I f that is the case, Then the transmitter sent 01 as the parity bits and there was one bit error, because we received 00. The path metric of the new state, PM['01',*i+1*] and PM['11', *i+1*], Finally we can construct the path metric as (1)

(1)

Where α and β are two predecessor states.

Finding the most likely path:

We can now describe hoe the decoder finds the most likely path. Initially, state '00' has a cost of o and the other 2^{k-1}-1 states have a cost of ∞. The main loop of the algorithm consists of two main steps: calculating the branch metric for the next set of parity bits, and computing the path metric for the next column. The path metric computation may be thought of as an add-compare-select procedure. Add the branch metric to the path metric for the old state, Compare the sums for paths arriving at the new state, select the path with the smallest value. The figure 6, represent the whole process of decoding.

5. Simulation Results

The figure 7 and figure 8 shows the RTL schematic diagram for the proposed design. This design reduces the power consumption 72% when compared with the existing shift register by using the single bit flip flop. The synthesis report for the design is also shown in the figure 9. Final report is also listed below for the design.

Fig. 7: RTL view of Viterbi decoder.

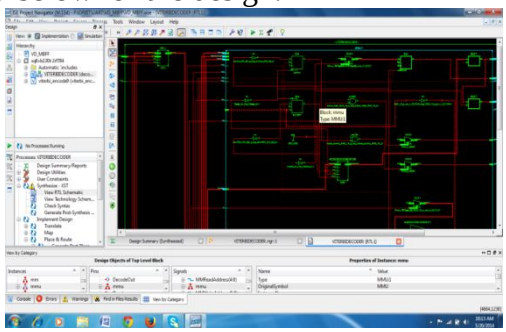

Fig. 8: RTL view of Viterbi decoder.

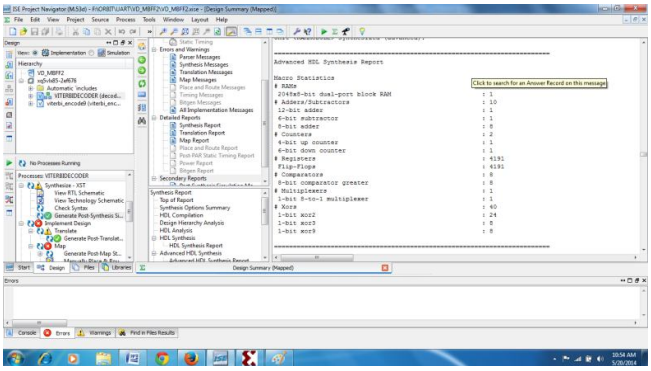

Fig. 9: Synthesis report of the design.

6. Conclusion

We have designed the convolution encoder and viterbi decoder by using a dynamic shift register which uses the multi bit flip flops. We also analyzed the timing analysis for the multi bi flip flop. The proposed design efficiently reduces the power consumption by 72% approximately. We also analyzed the timing for the whole design. We also shown the comparison tables for the existing single bit flip flop design and the proposed multi bit flip flop design.

References

[1] F. Chan and D. Haccoun, “Adaptive viterbi decoding of convolutional codes over memory less channels,” IEEE Trans. Commun. Vol. 45, no.11, pp. 1389-1400, nov.1997.

[2] V. Bhargava, D. Haccoun, R. Mathyas and P.NSP, digital communications by satellite.

[3] C. F. Lin and J. B. Anderson, " M-algorithm decoding of channel convolutional codes," presented at the Princeton Conf. Info. Sci. Syst.,Princeton, NJ, Mar. 1986

[4] D.Haccoun and M.Ferguson, Genralised stack algorithms for the decoding of convolutional codes" IEEE Trans. Inform. Theory, vol. IT-21, pp. 638-651, nov. 1975

[5] C.F.L in, "A truncated viterbi algorithm approach to trellis codes", Ph.D dissertation, Dep. Electrical eng.,, Rensselaer polytechnic Inst., Troy,NY, Sept. 1986."Bandwidth width modulations", Consultative

[6] K. S. Arunlal and Dr. S. A. Hariprasad" An efficient viterbi decoder" International Journal of Advanced

[7] Information Technology (IJAIT) Vol. 2, No.1, February 2012**.**

[8] R.A. Abdullah and N.R Shan hag," Error resilient low-power viterbi decoder architectures", IEEE trans. Signal process, vol. 57, n0. 12 pp. 4906-4917, Dec. 2009.

[9] J. B. Anderson and E. Offer, "Reduced-state sequence detection with convolutional codes," IEEE Trans. Inf. Theory , vol. 40, no. 3, pp. 965–972, May 1994.

[10] G. Prakash, K.Sathishkumar, B. Sakthibharathi, S. Saravanan, R. Vijaysai "Achieving reduced area by multi-bit flip flop design" 2013International Conference on Computer Communication and Informatics (ICCCI -2013), Jan. 04 – 06, 2013, Coimbatore, INDIA.