

FPGA Implementation of Viterbi Decoder Using Modified Register Exchange Method - An Overview

Surekha Shahare*, Dr. S.B.Pokle **

*(Dept. Of Electronics Engineering RCOEM, RTMNU University, Nagpur-13, India Email: surekha.shahare@gmail.com)

** (Professor & Head, Dept. of Electronics and communication engineering, RCOEM, Nagpur-13 India Email: hodec@rknec.edu)

ABSTRACT

Viterbi decoder employed in wireless technology are complex and dissipate large amount of power. High speed and small area are two important parameters in today's wireless technology. For decoding convolution code at the receiver end Viterbi decoder is being used. In this paper, design of Viterbi decoder by using modified register exchange method is used which reduces the amount of memory. Hence one pointer architecture of Viterbi decoder are used. A specification of convolutional code rate 1/3 with the constraint length 9 which reduces power consumption of an original Viterbi decoder by 55% and latency of 2 data bits. The proposed Viterbi decoder will be designed with VHDL Simulation will be done on Xilinx tool and implemented on Xilinx Spartan 2 based xc2s200pq208.

Keywords - Viterbi decoder (VD), Wireless, Low power, pointer, modified register exchange (RE).

I. Introduction

In 1967, Viterbi developed the Viterbi Algorithm (VA) as a method to decode convolutional codes [1]. The VA uses the trellis diagram to decode an input sequence. The Viterbi algorithm (VA) is an efficient method for the realization of maximum likelihood decoding of convolutional codes [2], [3]. The digital Viterbi decoder (VD) is used in wireless applications. Existing Viterbi Decoder are categorized by methodology of storing and retrieving the decoded data bits in and from the survivor memory unit (SMU); two methods are mainly used, the register exchange (RE) method and the trace back (TB) method [4]. When two or more paths end at the same state, the path with the smallest (or largest) path metric is selected as the most likely path [5]. Bit-serial approaches and operation reformulations have also been initiated [6], [7]. The RE method has been modified [8] to reduce its memory access rate. Their modified Register Exchange (MRE) method eliminates the Trace back operation and reduces the amount of memory. The MRE can be designed with memory or low latency, but no estimated power reduction figures were given. All of the previous design approaches were developed for low Constraint length Viterbi Decoder ($K=3$ to $K=7$). The decoders in [9], [10], and [11] were designed for wireless applications for which the constraint length must be $K=9$. Although Kang and Wilson have introduced a very low-power Trace back Viterbi Decoder [10], its speed is limited due to the use of sequential architectures for add-compare-select (ACS)

processing. The decoder that was work for lower power consumption and achieves speeds in the range of megabits per second [11]. Viterbi decoding has an advantage that it has fix decoding time and it is well suited to hardware decoder implementation [7]. The modified Register Exchange method [1], whose power consumption is further reduced in this work, utilizes the pointer concept that is widely used in software engineering. Instead of moving the contents of one row of memory to a second row of memory, the pointer to the first row is altered to point to the second row. The pointer to one row of memory simply carries the current state in the trellis of the Viterbi Decoder. The pointer implementation avoids the power hungry RE operations of the traditional Register Exchange method, and is referred to as pointer Viterbi decoder (PVD) [12] [13].

The Viterbi Decoder is composed of the three functional units in Fig 1.

1. The BM Unit (BMU) which calculates the BMs.
2. The Add Compare Select Unit (ACSU) which adds the BMs to the corresponding PMs, compares the new PMs, and then stores the selected PMs in the Path Metric Memory (PMM); at the same time, the ACSU stores the associated survivor path decisions in the Survivor Memory Unit (SMU).
3. The Survivor Memory Unit which stores the survivor path decisions.

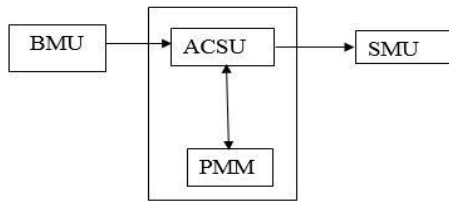


Fig 1. Basic Viterbi decoder

This paper is organized as follows. Basic introduction in Section 1. Reviews the Pointer Viterbi Decoder briefly and introduction of memory less version in section 2. Next, the proposed architecture of memory less Viterbi Decoder is presented in section 3, the Result is shown in section 4. Finally, Conclusion in Section 5.

II. Memoryless Viterbi Decoder

The Pointer Viterbi Decoder keeps path of the current row position of the decoder in the memory. It makes use of the fact that the bit appended to each row of memory is exactly the bit that is shifted into the pointer to form the new pointer to that row of memory. Each row of memory is used to trace the decoded bits, if an initial state is assumed. The first row of memory decodes the data, if an initial state, S0, is assumed. The last row records the decoded data, if an initial state, S3, is assumed, and so on. If the initial state is known, are all these rows of memory necessary? No. For example, if the initial state is zero, then only the first row of memory is needed. In other words, the storage of the decoded bits is necessary in order to choose only one row of memory at the end to represent the actual decoded bits. If the required row of memory is predetermined, then there is no need for the storage of the other rows. There is no need for the storage of the row that is assigned to the predetermined initial state, because the RE approach generates the decoded bits in the correct order. The decoded bits are produced, and then read out from the decoder. Thus, a memory-free Viterbi decoder can be implemented by solely resetting the encoder contents to zero for each bits that are encoded, where is the survivor path length. There is no need to interrupt the data sequence nor to transmit a long sequence of a zero data bits.

To show the functionality, a (four states) and rate convolutional encoder ((G0=101, G1=111 and G2=111) is employed to encode the input sequence of (10110100100). The code stream, (111, 011, 000, 100, 100, 000, 011, 111, 111, 011, 111) is generated and transmitted over a noisy channel.

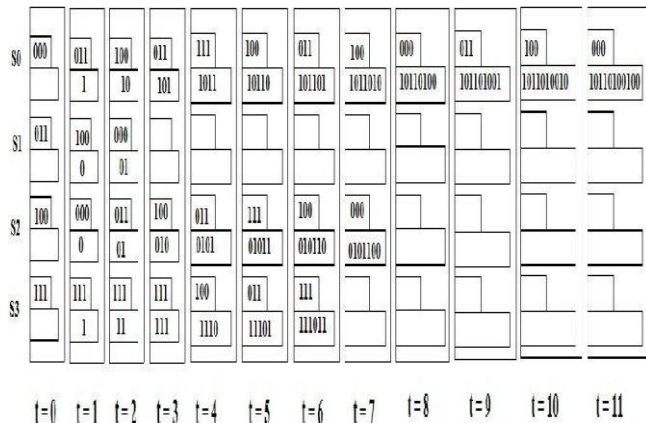


Fig. 2. MLVD approach with pointer implementation (the upper box carries the pointer and the lower box carries the decoded bits stored in memory).

The MLVD requires only one pointer to track the current position of the decoder in the trellis as shown in above Fig. 2. This pointer is reset to the initial state (zero) for each bits decoded. The MLVD is designed in VHDL for with the specifications listed in Table I

TABLE I
VD SPECIFICATION

Constraint length	K=9
Coding rate	r=1/3
Survivor path length	L=36
Generator polynomials	G0=101,G1=111,G2=111
Decision level	3 bit soft decision
Target speed	2 Mbps

III. Proposed Architecture of Memoryless Viterbi Decoder

The block Diagram of MLVD Design in VHDL as shown in fig.3

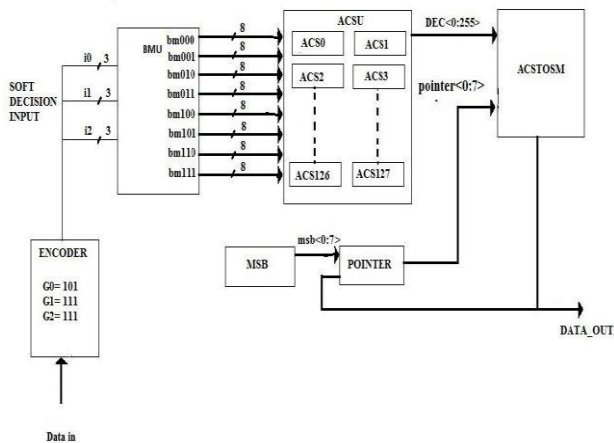


Fig.3. MLVD block Diagram

The next part is a discussion of the different parts of the MLVD design and their functionality.

3.1. Convolutional Encoder

The convolutional encoder that is implemented is for wireless applications. It is a K=9 and r=1/3 convolutional encoder shown in figure 4. To implement a 3-bit soft-decision Viterbi Decoder, the output of the encoder is translated from (0, 1) to (101, 011). 101 is the two's complement representation of the decimal number -3, and 011 is the representation of the decimal number 3. The output of the encoder is fed directly (without noise) into the first block of the Viterbi Decoder, the branch metric unit (BMU).

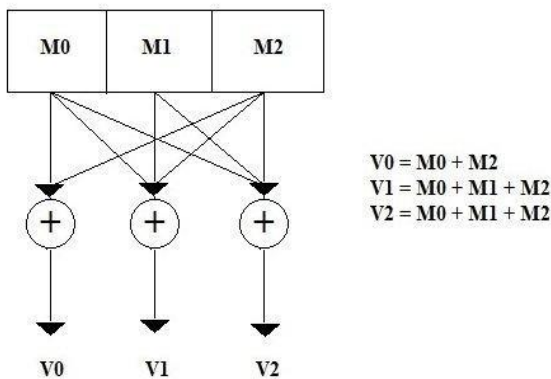


Fig.4. Rate 1/3 Convolutional Encoder

3.2. BMU

For binary convolutional codes, it is proven that linear distances (Hamming distances) can be used as the optimum branch metrics (BM) [15]. For three 3-bit soft decision input bits, (i0, i1, i2), each ranging from -3 to 3, eight 8-bit BMs are generated. The BMU performs simple add and subtract operations on the decision bits to generate the output as represented

in Fig. 5 [11], and the output of the BMU is still in a two's complement format, as shown in Fig. 5, then the bit serial format BM < 0:7 > is fed into the add-compare-select unit (ACSU).

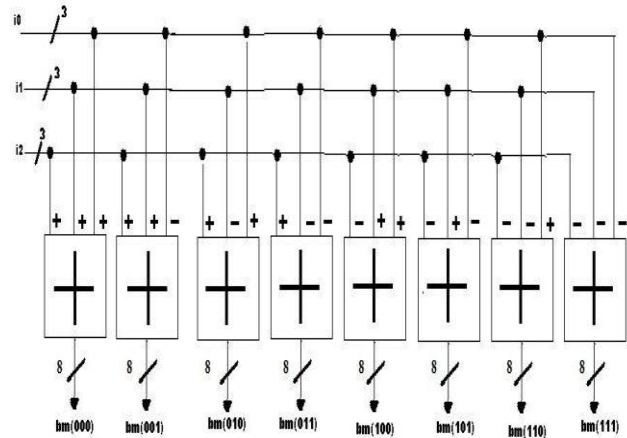


Fig 5. BMU operation

3.3. ACSU

The Add Compare Select Unit is composed of 127 units; each is composed of an ACS butterfly module, which adds the Branch Metrics to the corresponding path metrics (PMs), compares the new PMs, feeds the selected PMs Back into the ACSU, and generates some decision bits. Bit serial approach is used for the modified register exchange method. The bit serial ACS means that all the computations are done serially. The interconnections of state metrics among the ACS units as well as the interconnection of branch metrics between the BMU and the ACS units are all implemented with a single wire. Because of butterfly structure, Interconnections are reduced due to which power is reduced. The typical operation of one ACS butterfly module is as follow

$$\text{If } BM^{(i,p)} + PM^{(j,p)} < BM^{(j,p)} + PM^{(i,p)}$$

$$\text{Then Dec}^{(p)} = 0$$

$$PM^{(p)} = PM^{(i)} + BM^{(i,p)}$$

$$\text{If } BM^{(i,p)} + PM^{(j,p)} > BM^{(j,p)} + PM^{(i,p)}$$

$$\text{Then Dec}^{(p)} = 1$$

$$PM^{(p)} = PM^{(j)} + BM^{(j,p)}$$

$$\text{If } BM^{(i,q)} + PM^{(j,q)} < BM^{(j,q)} + PM^{(i,q)}$$

$$\text{Then Dec}^{(q)} = 0$$

$$PM^{(q)} = PM^{(i)} + BM^{(i,q)}$$

$$\text{If } BM^{(i)} + PM^{(i,q)} < BM^{(j)} + PM^{(j,q)}$$

Then $Dec^{(q)} = 1$

$$PM^{(q)} = PM^{(i)} + BM^{(j,q)}$$

According to the symmetric characteristics of the Viterbi Decoder

(Table I)

$$BM^{(i,q)} = BM^{(j,p)}$$

$$\text{And } BM^{(i,p)} = BM^{(j,q)}$$

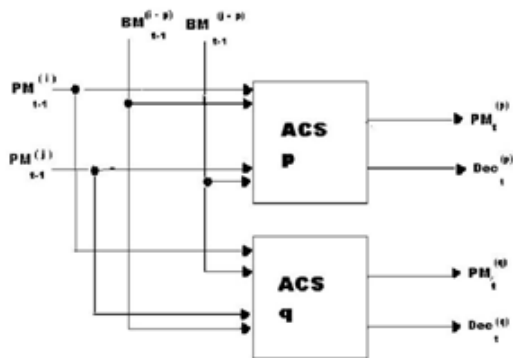


Fig 6. One ACS butterfly module

Therefore, only two BMs are connected to each butterfly unit as shown in Fig. 6. The bit serial approach proposed by Chang [11] is adopted for the MLVD implementation to reduce the routing overhead among the different ACSU modules.

3.4. ACSTOSM

The ACS to survivor memory (ACSTOSM) is employed to route the decision of the ACS module to the output. The ACSTOSM is a 256-to-1 decoder. The select signal for this large decoder is the output of the pointer module. The output of the ACSTOSM module is already the decoded output sequence of the MLVD, but is also fed back into the pointer module to update the current state in the decoding trellis.

3.5. Pointer

The pointer block contains the current state of the decoder (8 bits). It is reset to zero (the initial state of the encoder) for each of bits decoded. Then, for each bit decoded, the pointer content is updated, by the output of the ACSTOSM module. The exact position of the bit that will be updated is determined by the MSB block, which acts as a circular pointer to the pointer block. Power reduction estimation and performance evaluation are conducted in the next section.

3.6. MSB

The MSB module has eight outputs that point to the eight bits of the pointers in parallel. The reset signal causes the eighth bit of the MSB block to be high, which causes the eighth bit of any pointer to be the MSB of that pointer.

IV. Result

Viterbi Decoder are modelled using VHDL and Synthesized by Xilinx design manager FPGA logic. Some result are observed by using Xilinx tool. Simulation Result of Encoder and Branch Metric Unit shown in following figure. In Future implementation of ACS and Viterbi Decoder Module will be done..



Fig 7: Simulation result for Encoder

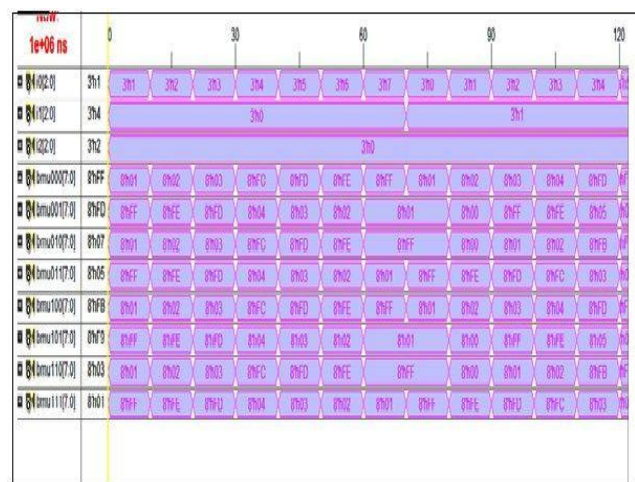


Fig 8: Simulation Result for Branch Metric Unit

V. Conclusion

Viterbi decoding is the best technique for decoding the convolutional codes. Viterbi decoder with constraint length 9 and code rate 1/3 has been developed. The main consideration is to decrease the

power dissipation. The two techniques used for decoding the convolutional codes are the Register Exchange method and Trace back method. TB method is used for larger constraint length but takes more time for decoding. RE method is simplest and fastest but more suitable for small constraint lengths. The Register Exchange method with large constraint length ($K=7$) is implemented with Modified Register Exchange Method. By using large Constraint length ($K=9$) we will reduce power and achieve better result than previous paper [2]. This implementation will reduce the power consumption to a large extent. The Add Compare Select unit and survivor memory management unit consumes most of the power. Bit serial approach is used for implementation of ACS unit. The amount of interconnections is reduced by using bit serial architecture and the butterfly concept unit which reduces the power consumption. Further, power is reduced by using the modified register exchange method. Modified register exchange method uses the pointer concept. The extra overheads are the registers required for storing the carry bits and the path metrics. The different modules are designed using VHDL and synthesized using Xilinx Integrated software environment (ISE). The design implementation will be done on Xilinx Spartan 2E. The Viterbi decoding is limited to lower constraint lengths. Viterbi decoder with Modified Register Exchange Method can be further investigated for higher constraint lengths.

REFERENCES

- [1] D. A. El-Dib and M. I. Elmasry, *Modified register-exchange Viterbi decoder for low-power wireless communications*, *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 2, pp. 371–378, Feb. 2004.
- [2] D. A. El-Dib and M. I. Elmasry, *Memory less Viterbi Decoder*, *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 12, Feb. 2005.
- [3] A.J. Viterbi, *Convolutional codes and their performance in communication systems*, *IEEE Trans. Commun.*, vol. COM-19, pp. 751–772, Oct., 1971.
- [4] G. Forney, *The Viterbi algorithm*, *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.
- [5] S.B. Wicker, *Error Control Systems for Digital Communication and Storage*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [6] H.-D. Lin and D. Messerschmitt, *Algorithms and architectures for concurrent Viterbi decoding*, *Proc. IEEE Int. Conf. Communications*, pp. 836–840, Jun. 1989.
- [7] B. Sklar, *Digital Communication Fundamentals and Applications*, Prentice Hall, Englewood Cliffs, New Jersey, 2000, Part 2 chapter 7.
- [8] K. Page and P. Chau, *Improved architectures for the add-compares-lect operation in long constraint length Viterbi decoding*, *IEEE J. Solid-State Circuits*, vol. 33, no. 1, pp. 151–155, Jan. 1998.
- [9] C.-Y. Tsui, R.-K. Cheng, and C. Ling, *Low-power ACS unit design for the Viterbi decoder*, in *Proc. IEEE Int. Symp. Circuits and Systems*, 1999, pp. 137–140.
- [10] J.-S. Han, T.-J. Kim, and C. Lee, *High performance Viterbi decoder using modified register exchange methods*, in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, May 2004, pp. 553–556.
- [11] J. K. Hinderling et al., *CDMA mobile station modem ASIC*, *IEEE J. Solid-State Circuits*, vol. 28, no. 3, pp. 253–260, Mar. 1993.
- [12] I. Kang and A. N. Wilson Jr., *Low power Viterbi decoder for cdma mobile terminals*, *IEEE J. Solid-State Circuits*, vol. 33, no. 3, pp. 473–482, Mar. 1998.
- [13] Y.-N. Chang, H. Suzuki, and K. Parhi, *A 2-mb/s 256-state 10-mw rate-1/3 Viterbi decoder*, *IEEE J. Solid-State Circuits*, vol. 35, no. 6, pp. 826–834, Jun. 2000.
- [14] Samir Kumar Ranpara, *A Low-Power Viterbi Decoder Design for Wireless Communications applications ASIC conference* Sept. 2005, Washington, D.C.
- [15] M. Kivioja, J. Isoaho and L. Vanska (2006). *Design and Implementation of Viterbi Decoder with FPGAs*. *Journal of VLSI*.
- [16] *Fpga Design and Implementation of low power array based Viterbi decoder circuit and system I: Regular paper on IEEE transaction 2005*.
- [17] *Low complexity based Viterbi decoder*” *Micro Nano mechatronics and Human Science 2007 IEEE transaction Symposium on June 2007*.
- [18] John G. Proakis (2001). *Digital Communication* Mc Graw Hill, Singapore.
- [19] Simon Haykin; *Digital communications*; Wiley, cop. 1988.
- [20] R. Henning and C. Chakraborty, *Low-power approach for decoding convolutional codes with adaptive Viterbi algorithm approximations*, *Proc. IEEE Int. Symp. Lower Power Electronics and Design*, pp. 68–71, Aug. 2002.
- [21] J. B. Anderson and S. Mohan, *Source and Channel Coding: An Algorithmic Approach*. Norwell, MA: Kluwer, 1991.
- [22] S.-S. Wang, *A state-reduction Viterbi decoder for convolutional code with large constraint lengths*, “*Master’s thesis, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.*”, Jun. 2002.
- [23] H.-L. Lou, “*Linear distances as branch metrics for Viterbi decoding of trellis codes*,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 6, Jun. 2000, pp. 3267–3270.
- [24] J. Ryu, S. Kim, J. Cho, H. Park, and Y. Chang, *Lower power Viterbi decoder architecture with a new clock-gating trace back unit*, in *Proc. 6th Int. Conf. VLSI CAD*, Oct. 1999, pp. 297–300.