International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-3 Number-3 Issue-12 September-2013

VHDL Implementation of Fast and Efficient Viterbi decoder

Rajesh. C[1], A Sreenivasa Murthy[2]

Abstract

Viterbi decoders are used in wide variety of communication applications. In this paper, we focus on different types of VHDL implementations of Viterbi decoder. The two approaches of Implementation of Viterbi decoder are register-exchange approach and trace back approach. There are two methods in trace back approach i.e. shift update and selective update. The behaviour of a Viterbi decoder is described in VHDL. A gate level circuit was obtained from the behavioural description through logic synthesis. We compared the performance characteristics of all approaches in terms of speed, area consumption, power and specific hardware components used by that particular design. Our experimental results show that the performance characteristics of selective update method are better compared to register-exchange and shift update method in terms of area and power consumption. In contrast, the performance characteristics of register-exchange method are better compared to selective update and shift update method in terms of speed.

Keywords

Register-exchange, trace back, shift update, Selective update

1. Introduction

A wireless cellular standard for CDMA (code division multiple access), IS95 employs convolutional coding. The hardware complexity of a Viterbi decoder is proportional to the number of states in the trellis, which is equal to the number of states of the corresponding convolutional encoder. For instance, the CDMA standard IS-95 employs a convolutional encoder with 8 states. Hence, a Viterbi decoder for the CDMA system has 256 states in the trellis, which results in complex hardware and high power dissipation unless designed properly for low power dissipation.

Rajesh.C, M.E, Electronics and Communication, UVCE, Bangalore University, Bangalore, India.

A. Sreenivasa Murthy, ECE Department, UVCE, Bangalore University, Bangalore, India.

The design of high performance Viterbi decoders has been investigated intensively in the past three decades [1], [5], [7], [10], [11], [12]. Recently, the low power design of Viterbi decoders has been an important issue for mobile and portable applications. Numerous techniques to reduce power dissipation have been proposed, and we review three recent works briefly. Chan, Lee, Lin and Chen implemented an adaptive Viterbi decoder based on an earlier work [2]. The adaptive decoder discards some states (in the trellis) with high path metrics dynamically during the decoding process. Seki et al. and Lang et al. suggested the use of a scarce state transition (SST) scheme [3]. The scheme employs a simple pre-decoder followed by a pre-encoder to minimize signal transitions at the input of a conventional Viterbi decoder, which leads to dynamic power dissipation. Kang and Wilson studied various issues in designing a low-power Viterbi decoder for the IS-95 CDMA system [5]. Their decoder employs various low-power design schemes such as state partition, gated control and gray coding. In this paper, we focus on different types of VHDL implementations of Viterbi decoder and analyse their performance characteristics.

2. Viterbi Decoder

In this section, we describe key parameters and the architecture of Viterbi decoders considered in the paper. We also explain two different methods for management of survivor path information and a block diagram of a Viterbi decoder.

2.1 Parameters of the proposed Viterbi decoder

The IS-95 CDMA system employs a 256 state convolutional encoder with a rate of r=1/2 in the forward link and convolutional encoder with the same number of states but with a reduced rate r=1/3 in the reverse link. In this paper, we consider a small-sized Viterbi decoder, which corresponds to a scaled-down version of IS-95 convolutional encoder in the reverse link, a 16 state encoder with a rate of r=1/3. The number of symbols in a frame for IS-95 is as high as 192, but we limit the number of symbols to 20 for our Viterbi decoder.

2.2 Management of survivor path information

Two basic approaches used to record survivor paths are register-exchange and trace back [1]. We describe the two approaches in detail.

In trace back, the survivor branch of each state is recorded. Using the one-bit information of each state, it is possible to trace back the survivor path starting from the final state (which is identical to initial state for the viterbi decoders considered in this paper).

Figure 1. describes the trace back approach [13]. For simplicity, only the significant bits of registers are shown in the figure. As in the register-exchange approach, a register is assigned to each state. Each register contains the past history of survivor branches, '1' for the upper branch and '0' for the lower branch in the figure. As the decoding process proceeds, new bits for survivor path information are appended to the existing one.

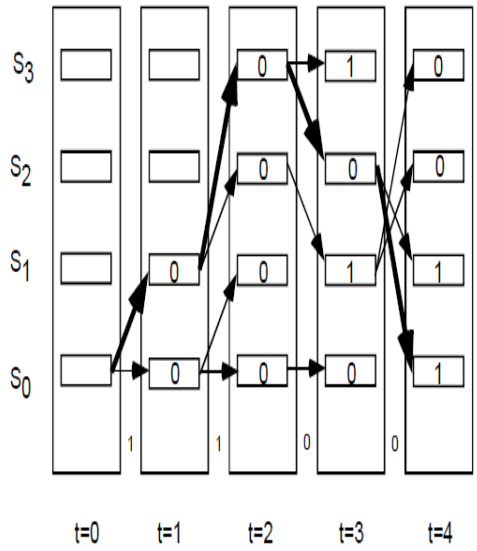

Figure 1: an example of trace back approach

In the register-exchange, a register assigned to each state contains information bits for the survivor path from the initial state to the current state. In fact, the register keeps the partially along the path, as illustrated in Figure 2 [13]. The register of state S_1 at t=3 contains '101', which is the decoded output sequence along the bold path from the initial state.

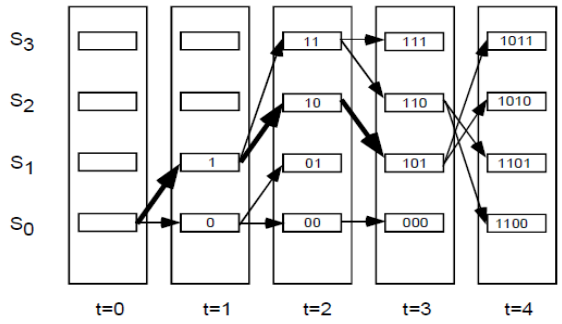

Figure 2: an example of register-exchange approach

The survivor path information is applied to the least significant bit of each register, and all the registers form a left shift operation at each stage to make room for next bits . Hence, each register fills in the survivor path information from the least significant bit towards the most significant bit. This scheme is called **shift update**. The shift update is simple in implementation but causes high switching activity due to the shift operation and, hence, results in high power dissipation.

2.3 System architecture

The major building blocks of a Viterbi decoder are shown in Figure 3. The role of each block is briefly described below.

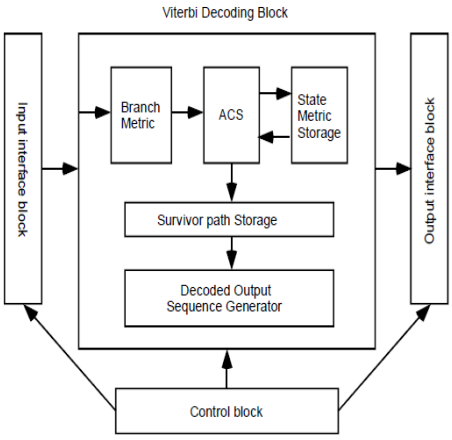

Figure 3 : Block diagram of a Viterbi decoder

Input and Output Interface: Input and Output interface blocks, provide the interface, including a serial to parallel conversion and vice versa.

Branch metric storage: The block calculates the branch metrics of each stage in the trellis, and a branch metric is calculated as the hamming distances between the received symbol and the expected symbol.

Path metric storage: The block stores the path metric of each state at the current stage.
ACS: The ACS (Add-Compare-Select) block is a collection of ACS units. An ACS unit receives two branch metrics and two path metrics. It adds each incoming branch metric to the corresponding path metric and compares the two results to select a smaller one. The path metric of the state is updated with the selected one. An ACS unit can be time-shared between multiple modules, but it incurs more power dissipation due to the control circuitry. One ACS unit is assigned to each state in our designs.

Survivor path storage: The block is a bank of registers, which record the survivor path of each state selected by the ACS module. A register is assigned to each state and the number of registers is equal to frame length (which is 20 in our Viterbi decoder)

Output generator: This block generates the decoded output sequence. It is trivial for the register-exchange approach since the decoded output is the content of the register of the final state. In the trace back approach, the block incorporates combinational logic, which traces back along the survivor path starting from the final state and generates the decoded output corresponding to the path.

3. Low power design

We explained two basic approaches used to record survivor paths: register-exchange and trace back. Both methods cause substantial switching activity and hence are inefficient in power dissipation. In this section, we discuss about a method through which we can reduce area consumption and in turn power dissipation.

In the trace back approach, one flip-flop is necessary to record the survivor branch for each state per stage. The shift update scheme forms a shift register for each state by collecting the flip-flops in the horizontal direction as shown in Figure 4 (b). The survivor branch information is filled into the least significant bits of registers. There is another method called **selective update**, in which registers are formed vertically as shown in Figure 4 (a). In this method, the survivor branch information is filled into registers from left to right as time progresses.

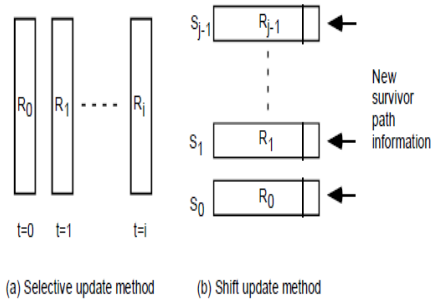

Figure 4: Shift update Versus Selective update

The key difference between two schemes is that the content of a register in the selective update method does not change once it is updated. Hence, the register incurs less switching activity, thus reducing power dissipation.

3.1 Toggle filtering of the Output generation block

The output generator block in the trace back approach traces back the survivor path after all the symbols have been received and generates the decoded output sequence. The block is a combinational circuit, which can be active during only one clock cycle as shown in Figure 5.

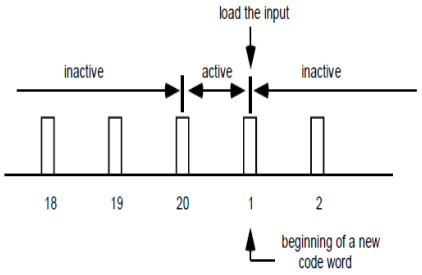

Figure 5: Activation of the output generator block

A block diagram of the output generation block is shown in Figure 6. Ignoring the AND gates with an enable input for time being, the block receives inputs from the survivor storage block containing the survivor path information. The block traces the survivor path at the end of the frame and generates the decoded output sequence. The decoded output sequence is loaded into a register at the first clock.

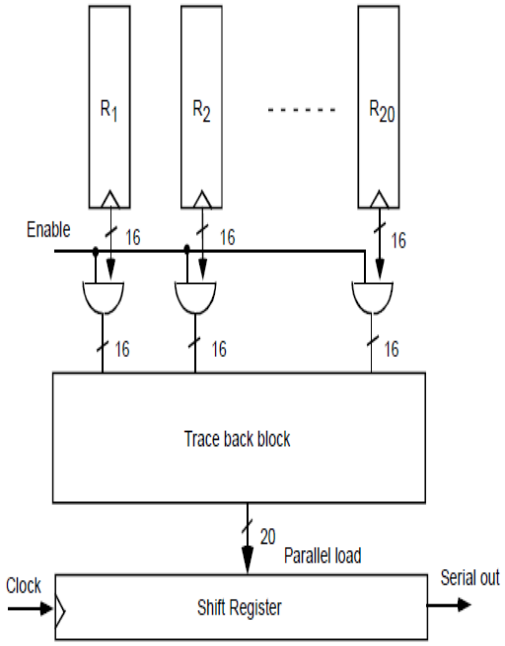

Figure 6: A block diagram of the output generator

Since the registers update the survivor path information progressively throughout the entire frame, the output generator receives spurious inputs, which causes unnecessary switching activity to dissipate power. The toggle filtering method blocks spurious inputs applied to the block. The array of AND gates and the enable signal shown in Figure 6 are introduced for this purpose. The enable signal is activated during one clock period at the end of frame as shown in Figure 6.

Since the registers update the survivor path information progressively throughout the entire frame, the output generator receives spurious inputs, which causes unnecessary switching activity to dissipate power. The toggle filtering method blocks spurious inputs applied to the block. The array of AND gates and the enable signal shown in Figure 6 are introduced for this purpose. The enable signal is activated during one clock period at the end of frame as shown in Figure 6.

4. Experimental Results

The behaviour of a Viterbi decoder is described in VHDL. A gate level circuit was obtained from the behavioural description through logic synthesis. We focussed on four different types of implementations of Viterbi decoder i.e. register-exchange approach, shift update with toggle filtering, selective update without toggle filtering and selective update with toggle filtering. We compared the performance characteristics of all four methods in terms of area consumption, speed and specific hardware components used by that particular design. In terms of area, comparison is in terms of slices, slice flip flops and input LUTs. In terms of speed, comparison includes Clock period, delay, Minimum clock period and minimum input arrival time before clock..In terms of power consumption, comparison is in terms of dynamic power. We have also studied the characteristics of static power. All comparisons with respect to above mentioned parameters are indicated in Table 1 to 4. The target device for our design is Xilinx xc3s400 FPGA device belonging to SPARTAN3 family with a speed grade of -5.

4.1 RTL Views

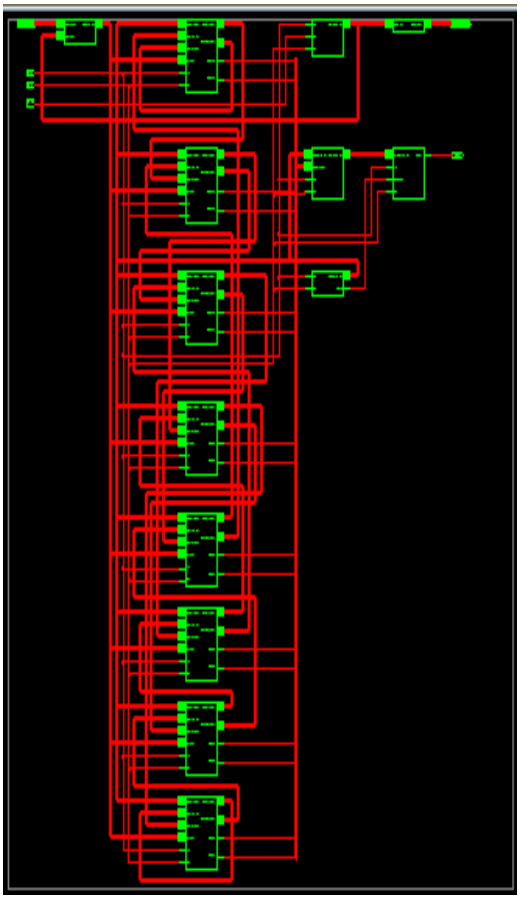

Figure 7: RTL Schematic of Viterbi Decoder

4.2 Comparison of all four methods in terms of area consumption, speed and specific hardware components used by that particular design

Table 1: Comparison of all four methods in terms of area consumption

Parameter	Register-exchange approach	Shift-update with toggle filtering	Selective update without toggle filtering	Selective update with toggle filtering
Number of slices	785	439	377	376
Number of Slice Flip-flops	446	394	360	360
Number of four input LUTs	1507	807	623	623

Table 2: Comparison of all four methods in terms of speed

Parameter	Register-exchange approach	Shift-update with toggle filtering	Selective update without toggle filtering	Selective update with toggle filtering
Minimum period	12.692 ns	25.239 ns	24.288 ns	24.288 ns
Clock period	12.692 ns	25.239 ns	24.288 ns	24.288 ns
Delay	12.692 ns(Levels of logic =8)	25.239 ns(Levels of logic =16)	24.288 ns(Levels of logic =15)	24.288 ns(Levels of logic =15)
Minimum input arrival time before clock	12.914 ns	14.597 ns	14.470 ns	14.470 ns

Table 3: Comparison of all four methods in terms of hardware components used

Parameter	Register-exchange approach	Shift-update with toggle filtering	Selective update without toggle filtering	Selective update with toggle filtering
Registers	446	397	360	360
Flip-flops	446	397	360	360
Bells	1665	1002	811	810
Total equivalent gate count for design	13,051	8,546	7,143	7,146

Table 4: Comparison of all three methods in terms of dynamic power

Comparison Parameter	Register-exchange approach	Shift-update method	Selective update method
Dynamic power	2.09 mW	1.55 mW	1.20 mW

4.3 Implementation Observations

The implementation of Viterbi decoder is illustrated in various pictorial views obtained during the process of realization i.e. Figure 7 shows the RTL Schematic of Viterbi decoder. Table 1 to 4 shows comparisons of all four methods in terms of different parameters.

5. Conclusions

This research work projects the design approach of Viterbi decoder. We have observed the simulation and synthesis results of all four approaches of Viterbi decoder. In terms of area and power consumption, we conclude that the selective update method is better compared to shift-update and register-exchange method. Therefore selective update method results in low power. In terms of speed characteristics, we conclude that register-exchange method is better compared to selective update and shift update method. This is because in register-exchange method, there is no trace back and hence we obtain decoded data faster. In terms of hardware components used, we conclude that the selective update method use less digital devices compared to shift update and register-exchange method.

Acknowledgement

The author would like to thank Dr. A Sreenivasa Murthy from UVCE, Bangalore University for his guidance and help.

References

[1] S.B Wicker, Error control systems for digital communications and storage, prentice hall, 1995.
[2] M-H Chan, W-T Lee, M-C Lin and L-G Chen, "IC design of an adaptive Viterbi decoder", IEEE Tans. On Consumer electronics, Vol.42, pp. 52-61, Feb 1996.
[3] K. Seki; S. Kubota; M. Mizoguchi and S. Kato, "Very Low power consumption Viterbi Decoder LSIC employing the SST (Scarce state transition) scheme for Multimedia Mobile communications", Electronics-Letters, IEE, Vol.30, no 8, pp.647-639, April 1994.
[4] L. Lang; C.Y Tsui and R.S Cheng and "Low Power Soft output Viterbi decoder scheme for turbo code decoding", Proc.int. Symp. On circuits and systems, pp.1369-1372, vol.2, 1997.
[5] Kang and A.N Wilson Jr, "Low- power Viterbi decoder for CDMA mobile terminals", Conference paper; Journal article, IEEE-Journal-of-solid-state-circuits. Vol.33, no.3, p.473-82, March 1998.
[6] G.K. Yeap, Practical Low Power Digital VLSI design, Kluwer Academic publishers, 1998.
[7] D. Garrett and M. Stan, "Low power architecture of the soft-output Viterbi algorithm", Electronic-Letters, Proceeding 98 for ISLEPD'98, pp.262-267, 1998.
[8] I. Kang; A.N. Wilson, Jr, "A low-power state-sequential Viterbi decoder for CDMA digital cellular applications", ISCAS 96. IEEE, New York, pp.272-275, vol.4 1996.
[9] S. Kubota, K. Ohtani and S. Kato, "A high-speed and high-coding-gain Viterbi decoder with low power consumption employing SST (Scarce state transition) scheme", Electronics Letters, 22(9), pp. 491-493, 1986.
[10] B.K. Min and N. Demassieux, "A versatile architecture for VLSI implementation of the Viterbi algorithm", in Proc. ICASSP, pp.1101-1104, May 1991.
[11] D. Oh and S. Hwang, "Design of a Viterbi decoder with low power using minimum transition trace back scheme", Electronic-Letters, IEE, Vol.32, No.22, pp.2198-2199, Oct.1996.
[12] P.J. Black and T.H. Meng, "A 140-Mb/s, 32-state, radix-4 Viterbi decoder", IEEE journal of Solid state circuits, vol. 27, pp. 1877-1885, Dec. 1992.
[13] www.ieeeexplore.ieee.org.

Rajesh. C was born in Bangalore, India on 31st January 1987 He received his B.E degree in Electronics and Communication from VTU, Karnataka, India in 2008. He is currently pursuing M.E in Electronics and Communication from Bangalore University, Bangalore, India.

Dr. A Sreenivasa Murthy is currently working as Associate Professor in UVCE, Bangalore University, Bangalore, India.