Design and Implementation of Low-Power Viterbi Decoder for Software-Defined WiMAX Receiver

Sherif Welsen Shaker, *Member, IEEE,* Salwa Hussien Elramly, *Senior Member, IEEE* and Khaled Ali Shehata, *Member, IEEE*

***Abstract* — Viterbi algorithm is employed in wireless communications to decode the convolutional codes; those codes are used in every robust digital communication system. Such decoders are complex and dissipate large amount of power. In Software Defined Radios (SDRs), Field Programmable Gate Array technology (FPGA) is considered a highly configurable option for implementing many sophisticated signal processing tasks. In this paper, a low-power, adaptive Viterbi decoder for WiMAX receiver is described using a VHDL code. The proposed design is implemented on Xilinx Virtex-II Pro, XC2vpx70 FPGA chip. FPGA Advantage Pro package provided by Mentor Graphics is used for VHDL description and ISE 10.1 by Xilinx is used for synthesization.**

***Keywords* — FPGA, Software Defined Radios, Viterbi Decoder, VHDL, WiMAX.**

I. Introduction

Many sophisticated signal processing tasks are performed in SDR that can be implemented on FPGA, including advanced compression algorithms, channel estimation, power control, forward error control, synchronization, equalization, and protocol management… etc [**1**], [**2**].

Channel coding may be considered as one of the most challenging signal-processing tasks performed in SDR. Most digital communication systems use convoulutionl coding to compensate for Additive White Gaussian noise (AWGN) and the effects of other data degradation like channel fading and quantization noise. Viterbi decoding of convolutional codes has been found to be efficient and robust for the advantage that it has a fixed decoding time and it well suites to hardware decoding implementation [**3**]. The complexity of the Viterbi decoder grows exponentially with the increase of the number of the states of the convolutional encoder which relates to the constraint length of the corresponding encoder. Therefore, careful designs of the decoder have to be found to realize such a practical decoder. The aim of any decoder realization might be to reduce the area, reduce the power consumption or to enhance the performance. While reducing the area can be gained by the advances reached in electronics, low power designs have to be developed especially because they are important issues for mobile and portable applications. This paper proposes low power, adaptive architecture for developing and modeling a Viterbi decoder for WiMAX software radio receiver. Puncturing/de-puncturing patterns are used in order to achieve code rates 2/3 and 3/4 that are higher than 1/2. The proposed design can be fabricated as a low power Viterbi decoder on ASIC for either a base station or portable unit.

II. Viterbi Algorithm

The Viterbi algorithm proposed by A.J. Viterbi is known as a maximum likelihood decoding algorithm for convolutional codes. So, it finds a branch in the code trellis most likely corresponds to the transmitted one. The algorithm is based on calculating the Hamming distance for every branch and the path that is most likely through the trellis will maximize that metric [**3**]. The algorithm reduces the complexity by eliminating the least likely path at each transmission stage. The path with the best metric is known as the survivor, while the other entering paths are non-survivors. If the best metric is shared by two or more paths, the survivor is selected from among the best paths at random.

The selection of survivors lies at the heart of the Viterbi algorithm and ensures that the algorithm terminates with the maximum likelihood path. The algorithm terminates when all of the nodes in the trellis have been labeled and their entering survivors are determined. We then go to the last node in the trellis and trace back through the trellis. At any given node, we can only continue backward on a path that survived upon entry into that node. Since each node has only one entering survivor, our trace-back operation always yields a unique path. This path is the maximum likelihood estimate that predicts the most likely transmitted sequence. [**4**].

Various coding schemes are used in wireless packet data network of different standards like GPRS, EDGE and WiMAX to maximize the channel capacity [**5**], [**6**]. WiMAX 802.16e, has the Viterbi with constraint length 7 and rate 1/2 with tail biting as mandatory and zero tail as optional [**7**]. This paper focuses on the implementation of

Sherif Welsen Shaker is with the Nanoelectronics Integrated Systems Center, NISC, Nile University, Giza, Egypt (phone: +20106444379; e-mail: swshaker@nileuniversity.edu.eg).

Salwa Hussien Elramly, is with Faculty of Engineering, Ain-Shams University, Cairo, Egypt; (e-mail: sramlye@netscape.net).

Khaled Ali Shehata, is with Arab Academy for Engineering and Technology, Cairo, Egypt; (e-mail: k_shehata@aast.edu).

the WiMAX Viterbi decoder with constraint length K = 7 and variable code rate r = 1/2, 2/3 and 3/4 convolutional encoder with connection polynomials g $(171, 133)_8$. An optimal trace back depth of 5 * K may be used as stated in [**4**]. We consider the number of symbols to be 40 per each frame. The tail sequence in each frame is fixed to 0's and resets the convolutional encoder to its initial state. Therefore, the trace-back operation can start from a known state.

III. Architecture of Viterbi Decoder

In this paper, we assume that the input to our proposed design is an identified code symbols and frames, i.e. the design decodes successive bit stream and the proposed decoder has no need to segment the received bit stream into n-bit blocks that are corresponding to a stage in the trellis in order to compute the branch metrics at any given point in time [**8**]. The architecture of the Viterbi decoder is illustrated in Fig. 1.

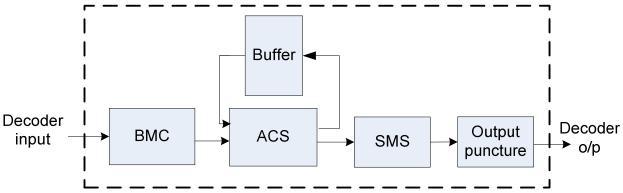

Fig. 1. Basic building blocks of the Viterbi decoder.

A. The Branch Metric Computer (BMC)

This is typically based on a look-up table containing the various bit metrics. The computer looks up the n-bit metrics associated with each branch and sums them to obtain the branch metric. The result is passed along to the path metric update and storage unit. The dashed rectangle in Fig. 2 shows the BMC [**9**].

B. The Path Metric Updating and Storage

This takes the branch metrics computed by the BMC and computes the partial path metrics at each node in the trellis. The surviving path at each node is identified, and the information-sequence updating and storage unit notified accordingly. Since the entire trellis is multiple images of the same simple element, a single circuit called Add-Compare-Select may be assigned to each trellis state.

C. Add-Compare-Select (ACS)

ACS is being used repeatedly in the decoder [**4**]. A separate ACS circuit can be dedicated to every element in the trellis, resulting in a fast, massively parallel implementation. For a given code with rate 1/n and total memory M, the number of ACS required to decode a received sequence of length L is $L\times2^M$. In our implementation we combined both the BMC and the ACS in one unit representing a single wing of each trellis butterfly as illustrated in Fig. 2.

D. Survivor Memory Management (SMM)

This is responsible for keeping track of the information bits associated with the surviving paths designated by the path metric updating and storage unit. There are two basic design approaches: Register Exchange and Trace Back. In both techniques, a shift register is associated with every trellis node throughout the decoding operation [**10**]. Since one of the major interests is the low power design, the proposed decoder in this paper has been implemented using the trace back approach which dissipates less power.

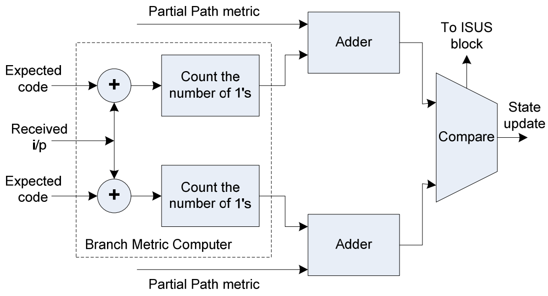

Fig. 2. ACS module.

Our proposed SMM is shown in Fig. 3, where for the encoder with constraint length K=7 and code rate 1/2, 40 registers are needed, each of them with size 64 bits. The survivor branch information is computed as following:

For the current state, if the input symbol is coming from the upper branch then set the corresponding state flip-flop to 1 in the 64-bit register. If the input symbol is coming from the lower branch then clear the state flip-flop. The bit in the corresponding flip-flop for the current state is being used associated with the current state itself to determine what the previous state was.

As time progresses, the survivor branch information is filled into the registers from left to right. The key issue is that the content of each register does not change as soon as it is updated. This is a very useful in our low power design, as we don't have to activate the registers after updating hence the switching activity reduces and of course a reduction in power dissipation can be reached. This can be realized by the clock gating-scheme [**11**]. As shown in Fig. 3, the clock of each register is enabled only when the register updates its survivor path information, while the clock of all other registers is disabled ensuring them to hold their states. The clock is gated by the information coming from a ring counter that tells what the current state is so far. The current state of the counter is as the same as the number of the received code symbols.

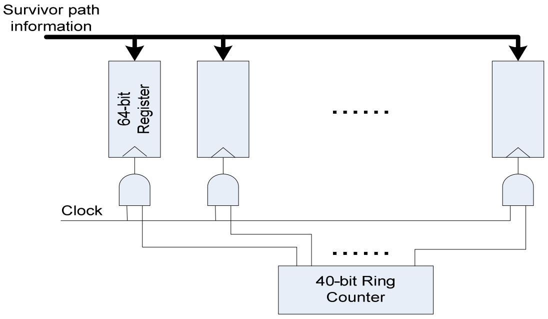

Fig. 3. Survivor Memory Management.

After receiving the entire frame, the trace-back unit in SMM traces-back the survivor path. The tracing back operation uses the bit in the corresponding flip-flop of the current state with the current state itself to determine the previous state **[12]**.

After tracing back the survivor path at the end of each frame, the decoded output sequence is being generated as the following:

For odd states, decide the input to the current state is symbol 1. For even states, decide the input to the current state is symbol 0. Since the decoding circuits have no need to work until all the frame symbols is being received, In our proposed design we enable those circuits one clock at the end of each frame, this saves more power. The survivor is then applied to a Parallel-to-Serial Interface that is implemented by 40-bit parallel-to-serial shift register.

E. Puncture/De-puncture

The output survivor sequence from the SMM block is applied to the de-puncture block that works in accordance with the puncture block in the convolutional encoder at the transmitter side. The puncture is implemented so that the decoder can work with rates 1/2, 2/3 and 3/4. Table 1 describes the puncturing method for K=7, r=1/2 convolutional code.

TABLE 1: PUNCTURING FOR CONVOLUTIONAL CODES.

Code rate	***r=1/2***	***r=2/3***	***r=3/4***
Parity 1(*x*)	11	10	101
Parity 2(*y*)	11	11	110
Output	x_1y_1	$x_1y_1\ y_2$	$x_1y_1\ y_2\ x_3$

IV. RESULTS

A Matlab code is first driven to evaluate the performance for the proposed Viterbi decoder and for verification the idea. Fig. 4 shows the BER curve vs. E_b/N_o using AWGN channel for both the uncoded BPSK and the convolutionaly coding with Viterbi decoding. The Matlab code is driven for hard decision decoding. Both the convolutional encoder with constraint length 7 and code rate 1/2 and our proposed Viterbi decoder are then described at the register transfer level by VHDL code, that consider the low power design technique using the concept of trace back approach along with clock gating. A VHDL code is also written for the Viterbi decoder with trace back and shift update approach to be compared in the power dissipation with the proposed design. The VHDL code includes the output de-puncture so that the decoder can operate with different rates such as 2/3 and 3/4 rather than code rate 1/2. The test-bench is written for both the encoder and the decoder as unique system, to provide the clock, control signals and the data to be encoded and then decoded. Noise generator module is implemented for the purpose of simulating the decoder; it simply adds noise by inverting some bits in transmitted frames. The test bench block diagram for the proposed decoder is shown in Fig. 5.

The simulation of the proposed Viterbi decoder has been made using Modelsim SE 6.4b digital simulator to test the function of the implemented decoder. Fig. 6 shows the simulation waveforms with applying an error pattern of 7 bit in error along the received frame. The result shows the efficient performance of the decoder in correcting up to 7 errors. After receiving the first frame, the data is kept decoded every clock cycle but delayed 40 clocks. This feature is very helpful in our design, since the trace back module does not work until the end of each frame and hence we don't have to activate the trace back module until the end of the frame and only for one clock cycle, by this feature, more saving in dissipated power can be reached.

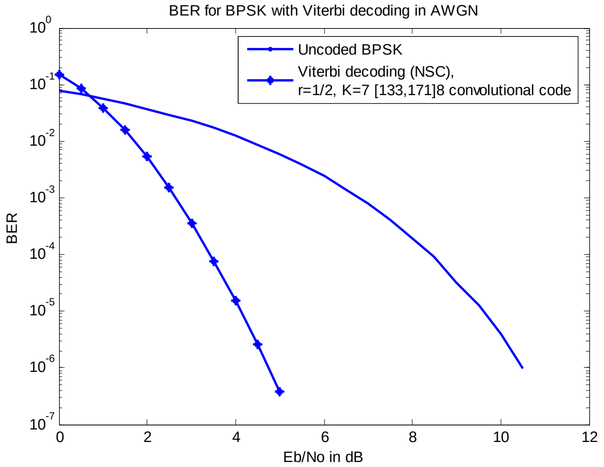

Fig. 4. Matlab simulation for Viterbi decoder.

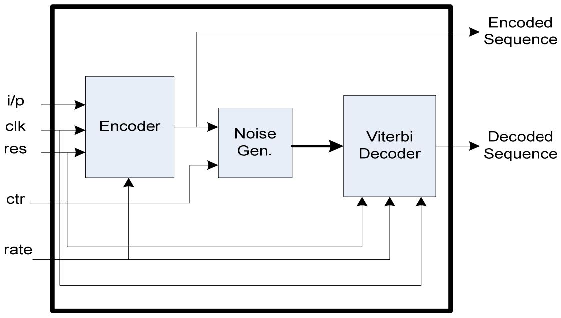

Fig. 5. Viterbi decoder, test bench.

Xilinx ISE 10.1 tools have been used for the synthesization purposes to map the design to the FPGA target technology. Xilinx Virtex-II Pro, xc2vpx70, with speed grade -6 has been selected; the design took about 8% of the total chip resources. Fig. 7 shows a part of the RTL schematic for the output shift register. It's obvious that the data is ANDed with the input "load" which is activated at the end of each frame, i.e. the data will not be loaded into the shift register until the end of the frame and this minimize the power dissipation. The results also showed that the proposed design can work with frequency up to 47 MHz. The proposed design and the one written using the shift update in trace back module are then applied to Xilinx XPower analyzer tool. Table 2 provides the power dissipation for both designs at clock frequency

12 MHZ and supply voltage Vccint=1.5V.

The results in table 2 shows how the dynamic power is being minimized for the proposed design using the trace back approach with clock gating and disabling the decoding circuits compared with other design that uses the shift update and does not benefits the gating. The results indicates the power reduction also when shifting the design on ASIC for using in base station.

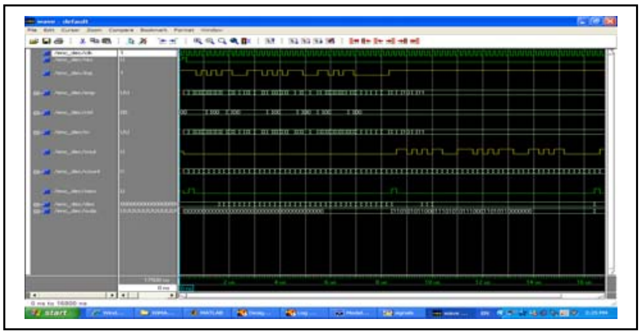

Fig. 6. Simulation of the decoder, 7-bit error pattern is corrected.

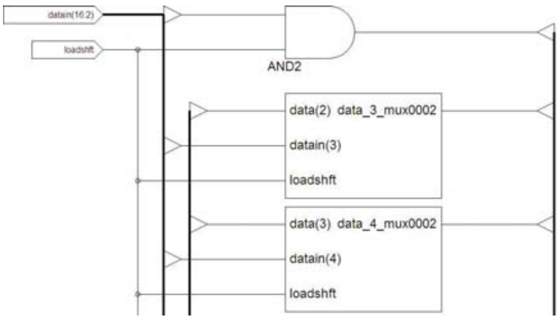

Fig. 7. RTL schematic of the output shift register

TABLE 2: COMPARISON OF POWER DISSIPATION.

	Design using Trace back with shift update	*Proposed Design*
Total Quiescent Power	0.185 W	0.054 W
Total Dynamic Power	0.061 W	0.014 W
Total Power	0.246 W	0.068 W

V. CONCLUSION

Features like higher flexibility, re-configurability and shorter time-to-market give the FPGA new opportunities for the effective insertion in SDR conditioning chain. In this paper, a design of low-power, adaptive Viterbi decoder for WiMAX receiver has been proposed. The design benefits from the concept of trace back with clock gating for power reduction. The design has been described by VHDL using FPGA Advantage Pro by Mentor Graphics, simulated using Modelsim SE 6.4b, and then targeted on Xilinx Virtex-II Pro, XC2vp70 FPGA using ISE design suit 10.1 by Xilinx. The power report for the proposed design have been driven using Xilinx XPower Analyzer tool, and it showed the reduction in the dynamic power dissipation to about 30% which is a good indication for power reduction when implementing the proposed design on ASIC. The design took about 8% of the total chip logic elements. The maximum operating frequency is 47 MHz, which is found adequate to our application.

REFERENCES

[1] M. Cummings and S. Huruyama, “FPGA in the Software Radio,” *IEEE Comm Magazine*, volume: 37, no. 2, pp. 108-112, February 1999.

[2] Xilinx Inc., Virtex-6 SXT for DSP and memory-intensive applications with low-power serial connectivity, http://www.xilinx.com/products/v6s6.htm. Last visited: April. 2009.

[3] A.J. Viterbi, “Convolutional codes and their performance in communication systems,” *IEEE Trans. Commun*., vol. COM-19, pp. 751-772, Oct., 1971.

[4] B. Sklar, “Digital Communication Fundamentals and Applications,” Prentice Hall, Englewood Cliffs, New Jersey, 2000, Part 2 chapter 7.

[5] GSM 05:03: “Channel coding”, *Version 8.9.0* Release 1999.

[6] GSM 03.64: “Overall description of the GPRS radio interface; Stage 2” *Version 8.12.0*. Release 1999.

[7] *IEEE Std 802.16e-2005*: “Part 16: Air interface for Fixed and Mobile Broadband Wireless Access Systems”.

[8] Sherif Welsen Shaker, Salwa Hussien Elramly, Khaled Ali Shehata, “FPGA Implementation of a Configurable Viterbi Decoder for Software Radio Receiver”, *IEEE 44th annual Systems Readiness Technology Conference, AUTOTESTCON 2009*, Anaheim, California, USA, 14-17 September 2009.

[9] Maunu, J.; Laiho, M.; Paasio, A., “Performance analysis of BMC and Decision Units in the differential analog Viterbi decoder”, *IEEE 25th Norchip Conference*, Aalborg, Denmark, Volume , Issue , 19-20 Nov. 2007 Page(s):1 – 4.

[10] Kamuf, M. Owall, V. Anderson, J.B., “Survivor Path Processing in Viterbi Decoders Using Register Exchange and Traceforward”, *IEEE Trans on Circuits and Systems II*: Express Briefs, vol. 54, Issue 6, pp. 537-541, June 2007.

[11] J. Oh and M. Pedram., “Gated clock routing for low-power microprocessor design”, *IEEE Transactions on Computer-Aided Design*, June 2001.

[12] Sherif Welsen Shaker, Salwa Hussien Elramly, Khaled Ali Shehata, “FPGA Implementation of a Reconfigurable Viterbi Decoder for WiMAX Receiver”. *IEEE 21st International Conference on Microelectronics, ICM 2009*, Marrakech, Morocco, 19-22 December 2009.