

FPGA Implementation of Soft Input Viterbi Decoder for CDMA2000 System

Miloš Pilipović, Marija Tadić, Faculty of Technical Sciences, Novi Sad
Mentor: dr Dragan Samardžija

Abstract — This paper presents a short overview of a soft input Viterbi decoder FPGA (Field-Programmable Gate Array) implementation for CDMA2000 (Code Division Multiple Access) wireless communication system in Verilog HDL (Hardware Description Language). The main goal of this project was resource-optimized implementation of the decoder on the target platform. It is well known that data transmissions over wireless channels are affected by attenuation, distortion, interference and noise, which affect the receiver's ability to receive correct information. Use of convolutional encoding with soft input Viterbi decoding is a powerful method for error-protection of transmitted information.

Keywords — CDMA2000, convolutional codes, FEC (Forward Error Correction), FPGA, soft input Viterbi decoder.

I. INTRODUCTION

IN wireless communication systems, error correction coding techniques play a very important role. Described in its simplest terms, error correcting coding involves the addition of redundancy to transmitted data to provide the means for detecting and correcting errors that inevitably occur in any real-world communication system. A Viterbi decoder uses the Viterbi algorithm for decoding a bit-stream that has been encoded using FEC (Forward Error Correction) based on convolutional codes. The Viterbi algorithm was conceived by Andrew James Viterbi, in April 1967, as an error-correction scheme for noisy digital communication links. It has found universal application in decoding the convolutional codes used in both CDMA and GSM digital cellular systems, dial-up modems, satellite, deep-space communications, IEEE 802.11 a/g, WiMAX, DAB/DVB, WCDMA [1]. It is now also commonly used in speech recognition, keyword spotting, computational linguistics, and bioinformatics.

The Viterbi algorithm is a MLP (*Maximum Likelihood Probability*) method, which means that the data output of the decoder, is the sequence that is most likely transmitted, given an observed received data sequence. There are other algorithms for decoding a convolutionally encoded bit

stream like for example, the Fano algorithm. Compared to the Fano algorithm, the Viterbi algorithm is the most resource consuming, but it has a significantly better performance.

There are several types of devices that can be used in a CDMA2000 communication system as illustrated in Fig. 1. This figure shows an overview of a CDMA2000 radio system [2]. It may include CDMA2000 multiple bandwidth radios and IS-95 CDMA compatible mobile telephones. The CDMA2000 mobile telephone devices are usually capable of operating as IS-95 CDMA and CDMA2000 mobile radios. This figure also shows that the CDMA2000 system can mix and combine three standard 1.25 MHz IS-95 channels into the 3.75 MHz CDMA2000 channel.

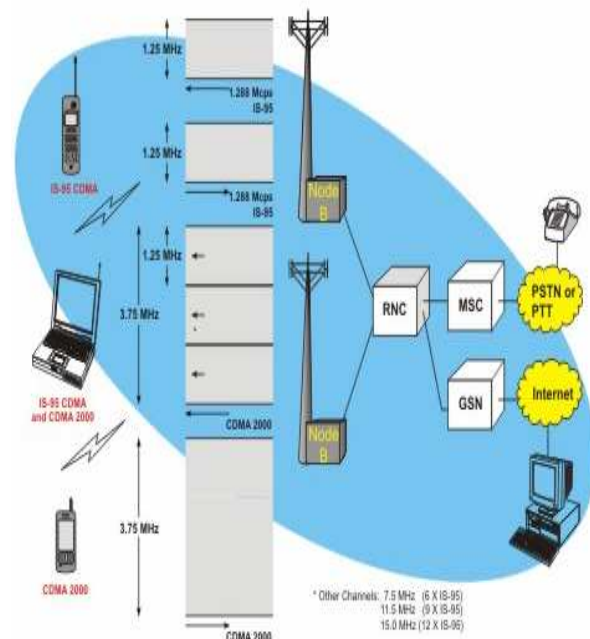


Fig. 1. Overview of a CDMA2000 communication system

In this paper we describe one implementation of a soft input Viterbi decoder on an FPGA platform. The FPGA DSP development board, *Altera StratixII EP2S60-F1020C4*, is used as the implementation platform.

Miloš Pilipović, Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia; (e-mail: milos.pilipovic@rt-sp.com).

Marija Tadić, Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia; (e-mail: marija.tadic@rt-sp.com).

II. VITERBI DECODING SYSTEM STRUCTURE

A structure and short overview of the basic Viterbi decoding system is illustrated in Fig. 2. This figure shows three basic elements of the Viterbi decoding system: convolutional encoder, communication channel and Viterbi decoder [3].

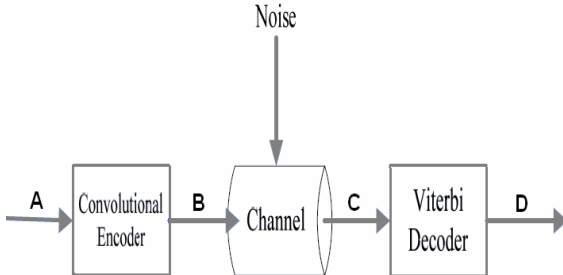


Fig. 2. A simple Viterbi decoding system

A. Convolutional Encoder

Convolutional encoder accepts input data bit-stream (A), while it generates coded output data bit-stream with added redundancy (B), also known as an error correction code. The presence of k storage elements in the encoder means that it acts as a FSM (finite state machine) with 2^k states. In fact, it is a Mealy machine, where the output is a function of the encoder current state and the current input. It is usually composed of shift registers and a network of XOR (Exclusive-OR) gates.

Fig. 3. shows convolutional encoder that corresponds to the implemented soft input Viterbi decoder described in this paper [4].

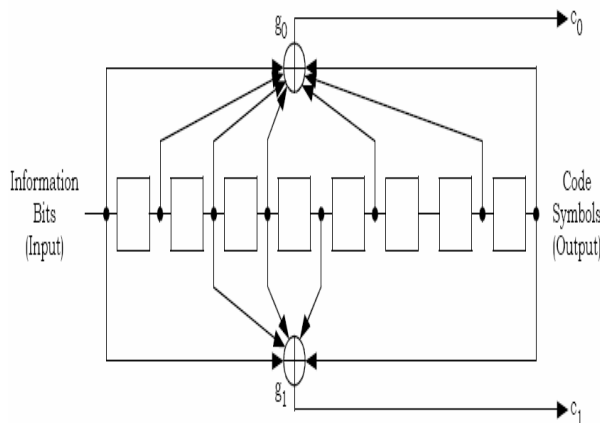


Fig. 3. Convolutional encoder $K = 9$, $r = 1/2$.

A convolutional encoder is represented by the number of output bits per each input bit (o), the number of input bits accepted at a time (i), the constraint length (K) and code generator functions also known as the generator polynomials (g_0 and g_1). The constraint length of the encoder indicates the number of bits stored in each shift register, including the current input bits. The code

generator functions specify how input bits contribute to the generated output. As illustrated in Fig. 3. the constraint lengths is $K = 9$ (i.e. number of shift registers is eight), while the code generator functions are $g_0 = 753$ (octal), and $g_1 = 561$ (octal). The convolutional encoder with the g_0 and g_1 generator polynomials, generates two code bits for each input data bit. The code rate $r = 1/2$ is defined as a ratio between the number of input bits and output bits i / o .

The convolutional encoder parameters are shown in Table 1.

TABLE 1: CONVOLUTIONAL ENCODER PARAMETERS.

| Symbol | Value |
|--------|--------------------|
| i | 1 |
| o | 2 |
| r | 1/2 |
| K | 9 |
| g_0 | 111101011 (binary) |
| g_1 | 101110001 (binary) |

B. Viterbi Decoder

When the received pair of encoded bits arrives at the decoder input, the decoder must infer, from what is received, the original data bit-stream. The received encoded version of the data may have been corrupted (C on Fig. 2.). The soft input Viterbi decoder process the received sequence C and generates an estimate D of the transmitted data.

Since an encoded pair of bits has different significance depending on the particular encoder state, the receiver must attach different significance to the corresponding received pair depending on which state it believes the encoder is in. Since the decoder has no knowledge of the encoder state, it has to set and test the corresponding hypothesis. For each possible state that the encoder may be in, the decoder generates a measure, i.e. a branch metric (BM) quantifying how far the received data is from the idealized error-free transmission. To complete the hypothesis testing, the BMs are combined with a likelihood measure, known as the state metric (SM) which is based upon the previously received data. The metric combinations are measures of the likelihood of each branch being correct. As each destination state has two routes into it, a more probable route will be the one with the higher likelihood metric. This higher likelihood is saved and becomes the SM for the given state in the following cycle. The index of the path that has the highest likelihood (i.e., the *local winner*) is recorded in the *survivor* memory, also known as trellis structure which represents the states of the convolutional encoder at different times. It is used to determine the most probable state-space path that the decoder went through and consequently it is used to derive the original data bit-stream by performing trace-back.

A comprehensive tutorial on the Viterbi algorithm is reported in [5].

III. VITERBI (SOFT INPUT) DECODER IMPLEMENTATION

Architectural parameters for the algorithm were obtained through MATLAB simulation on a workstation. The soft decoder input has a 8-bit resolution. The implementation flow of the decoder, as illustrated in Fig. 4. consists of several steps. Fig. 5. shows the Viterbi decoder block diagram.

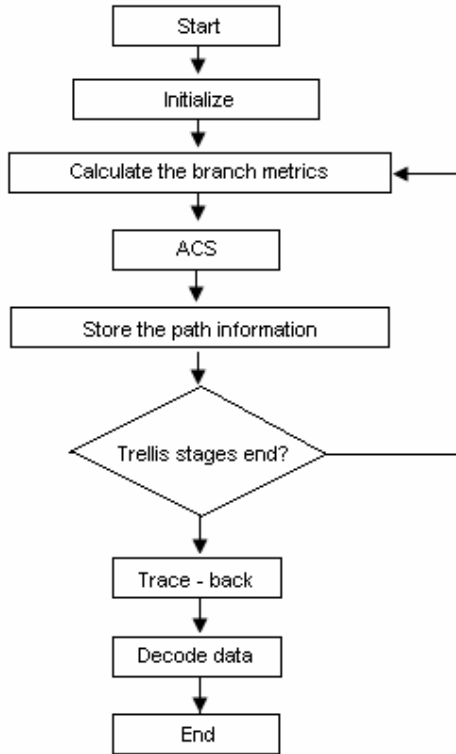


Fig. 4. Flow chart of the Viterbi decoder

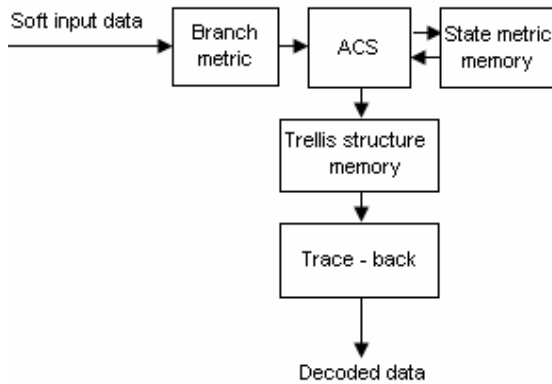


Fig. 5. Viterbi decoding block diagram

The soft input Viterbi decoder is realized as an FSM. Each step, as illustrated in Fig. 4. represents an FSM state in which decoder can be. The first step is the initialization of necessary algorithmic resources. In this step all decoder parameters, variables and memory required units are set up to their initial values. After that, the decoding process begins. As shown in Fig. 5. the Viterbi decoding block takes five major steps to finish the bit-stream decoding.

A branch metric section calculate the distance between the received sequence of soft decision values, and the branch word, which is the output from the convolutional encoder that results from the transition of one encoder state to another. For the 8-bit resolution, the soft decision values are in a range of -128 to +127, and are represented in 2's-complement format. Inputs to the decoder are blocks of 64 and 384 soft decision values. The branch word depends on the constraint length, the generator polynomials, and the code rate. The possible branch words are stored as a ROM table. This technique minimizes the hardware required to implement a large number of branch words. For soft-decision decoding the branch metric is the Euclidean distance. The Euclidean distance for code rate i/o is [6]:

$$LOCAL_DISTANCE = \sum_{n=0}^{o-1} [SD_n - G_n]^2 \quad (1)$$

which is further simplified to:

$$LOCAL_DISTANCE = \sum_{n=0}^{o-1} SD_n G_n \quad (2)$$

SD_n is soft-decision input, and G_n is the branch word for each path. For the code rate 1/2, the above equation becomes:

$$LOCAL_DISTANCE = SD_0 G_0 + SD_1 G_1 \quad (3)$$

The ACS (Add-Compare-Select) section receives the possible branch and the state metrics memory values. As design complexities increase, the use of vendor-specific IP (*Intellectual Property*) blocks has become a common design methodology. The state metrics memory and survivor path memory i.e. trellis structure, are implemented using manufacturer fully parameterizable, one-port RAM IP blocks, which are optimized for Altera device architectures, and consequently offer more efficient logic synthesis and device implementation [7]-[8]. The ACS section adds each incoming branch metric of the state to the corresponding state metric from state metric memory and compares the two results to select a larger one i.e. maximal metric wins (i.e. likelihood function). The ACS section at each unit of time gives information about the survived path of the two paths entering each trellis node. After that, it updates the state metric memory with selected value to be used in further processing. The ACS section repeats processing until the end of the trellis. The state metric memory stores partial path Euclidean distance. The trellis structure i.e. the survivor path memory section, records the survivor path of each state selected by the ACS section i.e. the index of the path that has the larger value i.e. highest likelihood.

After the whole trellis structure has been filled, the trace-back section begins by finding a maximum value from the state metric memory. The index of maximal value found, is the last state in trellis structure memory, from where the trace-back section begins. The trace-back determines the most probable path from the last state to the first state in the trellis structure memory i.e survivor path metric. Thereby it generates the decoded output sequence in the reversed order in a manner that all possible paths that end at half of the states indicate the

encoding of a one and those ending in the other half of the states indicate the encoding of a zero.

IV. IMPLEMENTATION RESULTS

An RTL level description of the soft input Viterbi decoder was written in Verilog HDL. The functional and timing simulation were performed using Cadence NCSim simulator software [9]. The results from NCSim simulator, matched with the results obtained through the corresponding MATLAB simulation. As the implementation tool, the QuartusII 6.0 design software is used with integrated modular compiler. As a result, the soft input Viterbi decoder uses 600 ALUTs (*Adaptive Look-Up Tables*) out of 48352 available ALUTs, 61696 memory bits of RAM (out of 2544192 total memory bits), the design does not use any of the available DSP resources, and maximum frequency is 215 MHz. This is all given for *StratixII EP2S60F1020C4* device.

An ALUT is the logic cell in the Stratix II device that is used as the output of logic synthesis. Two ALUTs comprise a single ALM (*Adaptive Logic Module*) which is the basic building block of Stratix II device.

The maximum frequency was obtained using the Altera timing analyzer tool. The maximum decoder throughput is 114,4 kbps.

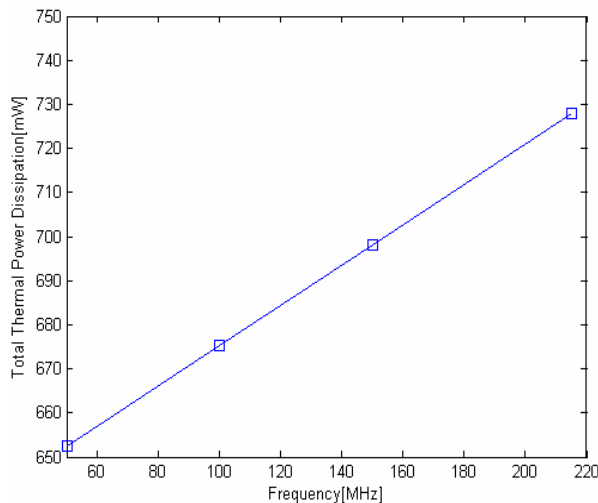


Fig. 6. Decoder thermal power dissipation

Fig. 6. shows a dependency between the decoder thermal power dissipation and working frequency. The above results were obtained using the Altera PowerPlay

power analyzer tool. It shows that total thermal power dissipation linearly increases with the operating frequency. The total thermal power dissipation of decoder for maximum decoder frequency of 215 MHz is 727,89 mW. Some of the selected operating conditions in the PowerPlay power analyzer tool are: ambient temperature is set to 25 °C, cooling solution is provided, junction-to-case is 0,13 °C/W, case-to-heat sink is 0,10 °C/W, heat sink-to-ambient is 1,8 °C/W, board thermal model is set-up to typical value.

V. CONCLUSION

In this paper we presented one implementation of the soft input Viterbi decoder. We showed that it can be efficiently implemented on the commercially available FPGA platform. The proposed solution was shown to be particularly efficient in terms of the required implementation resources and power requirements. There is a number of possible future directions that may be pursued. For example, methods to achieve higher levels of design parallelism and consequently higher throughputs are of interest. Furthermore, the decoder performance for different channel conditions and the decoder soft input resolution are to be investigated.

ACKNOWLEDGEMENTS

The work described above was partially supported by the 12004 project, year of 2008, of Ministry of Science of the Republic of Serbia.

REFERENCES

- [1] A. Viterbi, "Error bounds for convolutional codes and asymptotically optimum decoding algorithm," *IEEE Transactions on Information theory*, vol. It-13, no. 2, pp. 260–269, April 1967.
- [2] L. Harte, "Introduction to Code Division Multiple Access (CDMA)," Althos, August 2004.
- [3] "RTL implementation of Viterbi decoder," Dept. of Computer Engineering at Linkpings universitet, June 2006.
- [4] *Physical Layer Standard for cdma2000 Spread Spectrum Systems*, TIA/EIA INTERIM Standard, May 2002.
- [5] G. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, March 1973.
- [6] S. Shahzad Shah, S. Yaqub, and F. Suleman, "Self-correcting codes conquer noise," February 2001.
- [7] *Altera Stratix II EP2S60 DSP Development Board Data Sheet*, Altera Corporation 2005.
- [8] *Altera RAM Megafunction User Guide*, Altera Corporation 2007.
- [9] *Cadence SimVision User Guide*, Cadence Design Systems Inc., 2004.