Mathematical Problems of Computer Science 35, 137--139, 2011

Compact High Performance Algorithm for Convolutional Decoder

Arsen Hakhoumian, Tigran Zakaryan, Aramazd Muzhikyan and Vahan Nikoghosyan

Institute of Radiophysics and Electronics, Armenian National Ac.Sci.
Alikhanian 1, Asbtarak, 0203, Armenia

Abstract:

An efficient algorithm for convolutional decoder running in the mode of soft decision (1/2, K=7, 3-bit decision) in a FPGA Virtex-II is proposed. The results show that although increasing the length of the cutoff buffer improves the signal to noise ratio by 2dB, but it can significantly increase calculation time, which exponentially depends on the length of the cutoff buffer. At the optimal length of the cutoff buffer equal to 7, the obtained throughput is estimated up to 14Mbps.

1. Introduction

Modern telecommunication facilities that use traditional digital modulation techniques are widely using methods of channel coding, which allow improving the reliability of received symbols. In particular, various convolutional coding methods are in use, of which the most widely used now is Viterbi algorithm. Increased demands for performance and capacity of mobile networks have necessitated the creation of relatively cheap and at the same time effective hardware, implementing the process of channel coding, which allows the integration of communications devices into mobile. Specific requirements for mobile hardware, in turn, also determine the increased demands on the software [1-3], which is in fact an integral part of the hardware.

2. Description and Discussion of the Algorithm

This algorithm is specially designed for a programmable gate array, and its severe resource constraints will determine its structure. The algorithm was implemented by means of LabVIEW FPGA module. As we know, 7-bit Viterbi coding (K = 7 and the encoding rate 1/2) can utilize the scheme 171x133 (in octal notation). The decoder, respectively, has $K-1=6$ input registers, so that the number of its possible states is $2^{K-1}=64$. On input signal bits, the encoder can only go to one of two possible states, issuing two output signal bits. Accordingly, the encoder can proceed to any possible state only from the previous two states, depending on the value of the signal bits.

With a soft decoding scheme, signals to the input of decoder are fed as 3-bit signals with values ranging from -4 (signal "1") to +3 (signal "0"). To compute the Euclidean distance corresponding to each possible transition from this state to next ones, they are summed with the signs taken from the table, pre-compiled for each state. Transition to any possible state of the decoder is only possible from the other two states. The Euclidean distance for these two transitions of the received signal is directly calculated. Values of Euclidean distance metrics are added to the values of the two relevant paths that lead to this state from which it selects the one for which the value of the metrics is smaller. For equal values of path metrics, the decision is made randomly

In order to improve efficiency and increase performance, all the calculations for the opportunities are organized in parallel, such as calculation of metrics for each of the 64 possible ways. It is not necessary also to use the table of signs used in the calculation of Euclidean distances: they are already present in the appropriate branch of the algorithm for any possible way.

Since the notation for each possible trace in transition state of the decoder is a sequence of binary digits, it was suggested to use integers to represent each path in the algorithm instead of a bit array. For concatenation to this sequence of zeros or ones, we need only to double the path representing number (addition of "0") or add 1 to twice the number (addition of "1") by logical shift operator. We should initially set the value of all paths equal to 1 (i.e. introduce "non-working" bit) in order to make it possible the doubling. This bit is present in all subsequent operations, but is not taken into account.

Path records actually represent the trellis diagram of the decoder. The number of representing bits of path numbers determines the trellis length. During bit arrival, permanent checking of trellis length is made: $N_i \le 2^n$, where N_i is integer representation of the i-th path, and n is the maximum lattice length. (Binary representation of 2^n is one with n zeros. Since the path record is formed by adding bits to this one, then for any sequence of 0 and 1, N_i is greater or equal than 2^n, when the number of bits is equal to n).

If the length of the lattice is less than some value, the metric of all paths and the integers representing the respective paths are passed to the beginning of the next iteration to receive the next signal bit.

Otherwise, when the length of the lattice reaches a limiting value, the numbers representing the paths are compared with 2^{n-1} by "AND" Boolean function. This allows to find out the value of bit (0 or 1), standing in second left position (or in the n-th right position) of the path representing numbers. In each of these numbers this bit is the "oldest" one in the chain of binary symbols that form the trellis diagram, since we have introduced initialization ("non-working") bit on the most right position. In order to exclude them from the diagram, we subtracted 2^n from all path representing numbers (Fig. 1). The residue of the subtraction operation is a stripped-down trellis diagram, which is looped-back as numbers to the next iteration of the decoder to receive the next signal bit. The algorithm selects from few bits the appropriate one corresponding to the maximum value of the metrics and its value is transferred to the output of the decoder as the received bit. To limit the value of metrics and to prevent the overflow of registers, on each iteration of the values of all metrics the metrics value of one state, which remains the average level of metrics near zero value, is subtracted.

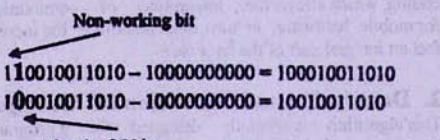

Рис. 1. Exclusion of the "oldest" bit

At each transition to the next iteration, the metrics values of different paths of trellis, the respective path numbers and their compliance with the existing states of registers must be considered. As mentioned above, the encoder can change the state only from the previous two states, depending on the input bit values. Status register of the encoder is represented by 6 bits (64 possible states). The input bit is concatenated to the left of this representation and the right less significant bit is removed. Hence, at "0" bit arrival, the coder can jump to this i-th state from $2i$ or $2i+1$ states (for states $0 \le i \le 31$), while at "1" bit arrival, the coder can jump to the same i-th state from $2(i-32)$ or $2(i-32)+1$ (for states $32 \le i \le 63$) (Fig.2). These expressions actually are those rules, which determine how the metrics and record of the paths are transmitted from considered iteration to the next.

The results show that although increasing the length of the cutoff buffer improves the signal to noise ratio by 2dB, but it can significantly increase calculation time, which exponentially depends on

1 + 101000 (40) → 110100 (52)
1 + 101001 (41) → 110100 (52)
0 + 101000 (40) → 010100 (20)
0 + 101001 (41) → 010100 (20)

Fig.2. Change of internal register state at new symbol arrival (decimal values)

the length of the cutoff buffer. Figure 3 shows the plots of bit error rate for different lengths of the cut-off buffer. At the optimal length of the cutoff buffer equal to 7, the obtained throughput is estimated up to 14Mbps.

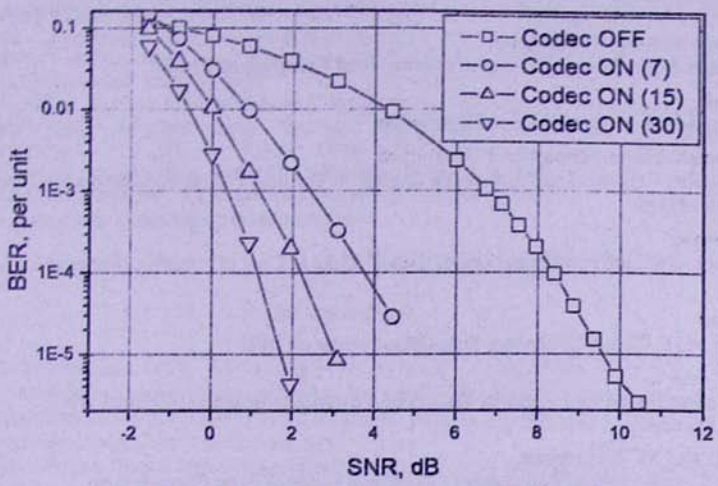

Fig. 3. BER vs SNR for different lengths of cut-off buffer

References

[1] M Hosemann, R.Habendorf and G.Fettweis, "Hardware-software codesign of A 14.4mbit - 64 State viterbi decoder for an application-apecific digital signal processor", *IEEE Workshop on Signal Processing Systems*, pp. 45-50, 2003.

[2] M.Röder and R.Hamzaoui, "Fast tree-trellis list viterbi decoding", *IEEE Transactions on Communications*, vol. 54, no. 3, pp. 453-461, 2006

[3] J.Campos and R.Cumplido, "A runtime reconfigurable architecture for viterbi decoding", *3rd International Conference on Electrical and Electronics Engineering*, pp. 1-4, 2006.

Բարձր կատարողականությամբ փաթույթային ապակոդավորման սեղմ ալգորիթմ

Ա. Հախումյան, Տ. Զաքարյան, Ա. Մուժիկյան և Վ. Նիկողոսյան

Ամփոփում

Առաջարկվում է արդյունավետ ապակոդավորիչի փաթույթային ալգորիթմ, որն աշխատում է որոշում կայացնելու փափուկ ռեժիմում (1/2, K=7, որոշում կայացնելու 3 բիթ) FPGA Virtex-II համակարգում: Ստացված արդյունքները ցույց են տալիս, որ կտրման բուֆերի երկարության կրկնակի մեծացումը չնայած և հանգեցնում է ազդանշան/աղմուկ հարաբերության 2 դԲ-ով լավացմանը, կարող է էականորեն մեծացնել հաշվարկի ժամանակը, որը ցուցչային կախվածություն ունի կտրման բուֆերի երկարությունից: K=7 օպտիմալ երկարության համար ստացվում է 14 Մբ/վ առավելագույն ունակությունը: