International Journal of Soft Computing 10 (6): 400-404, 2015
ISSN: 1816-9503

Universal Coding-Lossless Compression

V.A. Shurigin, V.V. Makarov, A.B. Vavrenyuk, D.M. Mikhaylov and M.I. Froimson
National Research Nuclear University "MEPhI" (Moscow Engineering Physics Institute),
Kashirskoe Highway, 31, 115409 Moscow, Russian Federation

Abstract: This study deals with the method for binary data lossless compression within the unknown statistics of message source. The method helps to restore the data binary chain dead true. The application of the method helps to significantly save the memory volume of the data storage devices to increase the data line capacity, etc.

Key words: Binary data compression, lossless compression, coding with three-digit groups of bit combinations, universal coding, statistics

INTRODUCTION

In the epoch of the total computerization with different computing devices, it is necessary to transfer, store and process the huge amount of data. Thus, to transfer these data it is necessary to compress them. But compressing can result in worse data quality, its loss or distortion (Tan and Wang, 2009).

The problem of data compression raises interest of many scientists (Zhang *et al*., 2015). The ides presented in scientific papers have their advantages and disadvantages. Researchers of the study have developed their own method of compression which is much more effective in comparison to its analogues.

In the present study, the method of universal coding with three-digit groups of bit combinations (TC) is introduced. The study, researchers developed this method. According to the definition, universal coding is defined by statistical excess in the sequence received after coding and tends towards zero while block length increases on which the source binary sequence is divided (Alexandrovich *et al*., 2011).

MATERIALS AND METHODS

The core of the method is a separation of the original sequence of binary data into blocks with n-capacity. Further, each block is assigned to three parameters calculated based on the block contents:

- Number of units in the block is k
- Number of their positions sum is s
- Number of each combination in the corresponding class (K and S set intersection element)-b(n, k, s)

Researchers have proved that R_n superabundance in the stream of the outgoing data goes to zero with n original blocks length growth, means coding is asymptotically optimal, i.e.:

$\lim R_n = 0$ where $n \to \infty$
R_n-coding superabundance: $Rn = \sup R_n(p)$ where $0<p<1$
$R_n(p) = n_{av}/n - H(p)$ where, n_{av} is an average length of the code word,
$H(p) = -(p\log_2 p + q \log_2 q)$-source entropy

When an additional partition value is inserted into TC a number of S block units sum positions (instead of the sum itself), more smooth going to zero is seen with blocks length growth, less labor intensity is needed, more affectivity for sources with possibility of appearing units <0.1 (or exceeding 0.9).

Labor intensity, determined by (according to coding theory) equal memory volume and the number of sum operations needed for TC realization and is described with power law of source blocks length ($1 \le n \le 3$).

Figure 1 shows the dependence of compression coefficient diagram from source block length for different p-values. In fact, source blocks compression coefficient with 128 bits length is between 2 and 20 for possibility of appearing of units between 0.1 and 0.001. Here, 10^4 bits are needed for realization. Modern development if the field

Corresponding Author: V.A. Shurigin, National Research Nuclear University "MEPhI" (Moscow Engineering Physics Institute), Kashirskoe Highway, 31, 115409 Moscow, Russian Federation

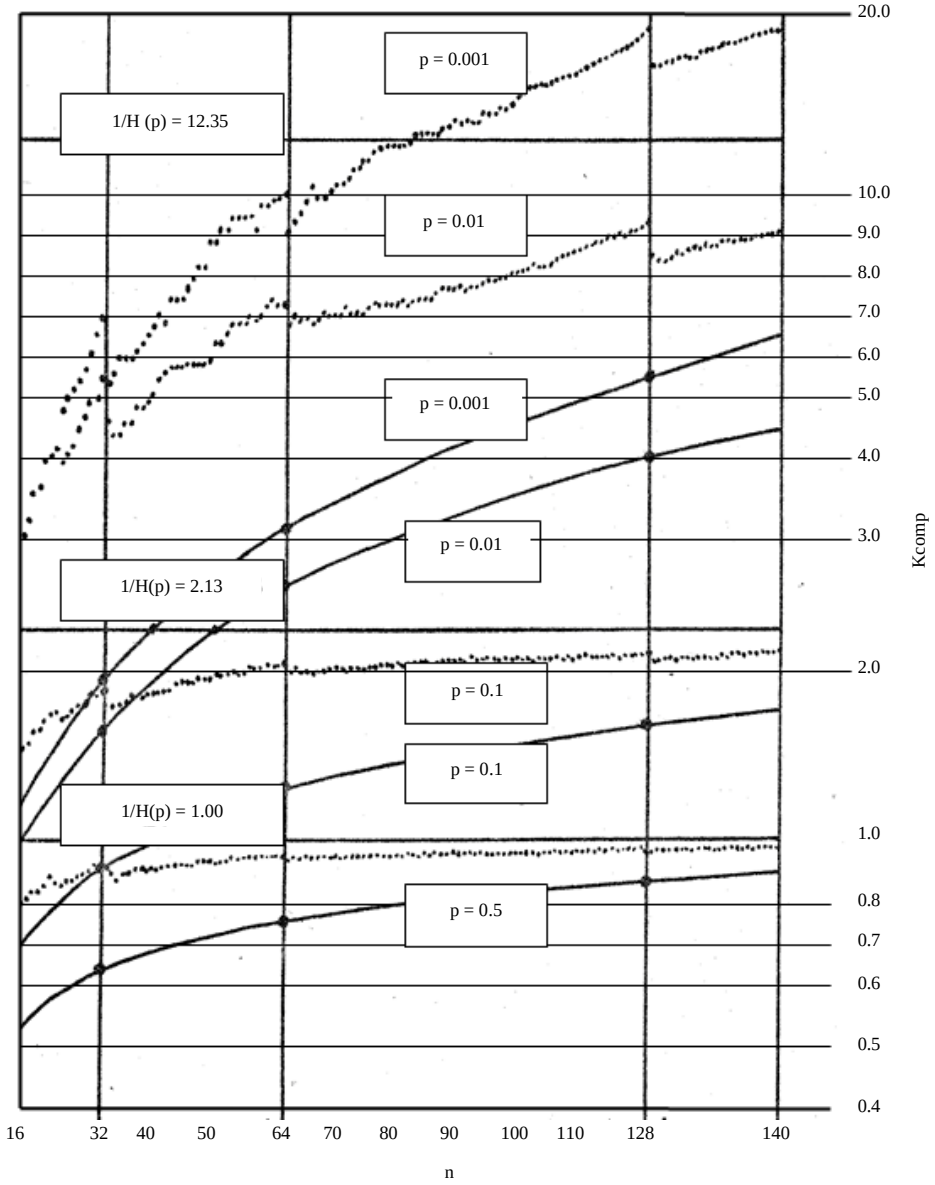

Fig. 1: Dependence of n compression coefficient for different p-values

of microelectronics and integration: flash-memory chips, one-crystal Electronic Computing Machine (ECM), usage of nanotechnologies both help to implement separation of 10000 bits into blocks (impossible before). With the "fastest", Table-like Realization Method -4.8 gigabytes are required which is equal to the memory volume of the modern flash card and compression is in the amount between 2 and 80. This value is practically the theoretical maximum.

Additional labor intensity lowering is achieved taking into consideration non-monotonicity of curve behavior in Fig. 1 (local areas exist on which ngrowth is not followed by κcomp (compression) growth) as well as adaptation installation in the coding procedure. During the coding procedure preliminary analysis of the growth of unit number in the present block and the sum of their positions is made. On the basis of the analysis coding is performed or block is transferred without any changes with the corresponding characteristics. The following TC realization variants are possible:

- Program approach with the help of ECM
- Software-hardware method with the help of single-board computer, e.g., "Gamstix Technology" of one company from California (chewing gum sized) or with the help of ECM in one crystal, e.g., by INTEL (single-chip) based on Intel XScale processor and Strata Flash memory

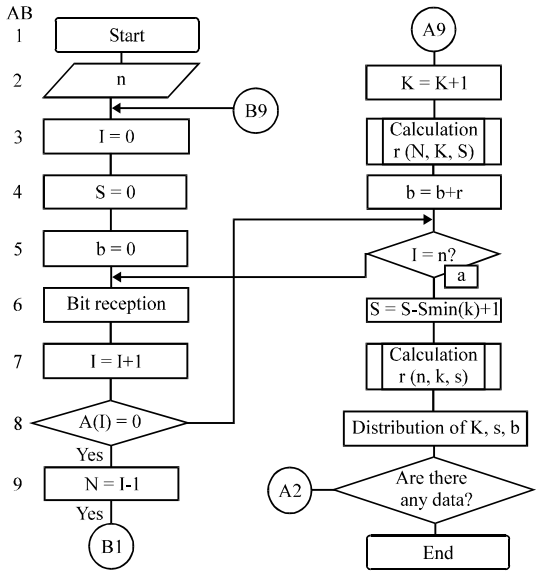

Fig. 2: Coding algorithm

- Development of the special device based on original microcircuit creation with flash technologies or adaptation of one of the serial flash-memory microcircuits

Researchers consider the third approach the most perspective. It would have been very attractive to create a microcircuit on which entry binary sequence with statical superabundance is situated and sequence without superabundance is taken off from its terminal, i.e., data compression is provided. In Fig. 2 TC realizing algorithm is provided. The following tags are used:

- k: number of units in the source block
- S: number of source block units positions sum
- b: number of certain code combination for k and S fixed values
- r (n, k, S): number of binary combinations which have k units and S sum of their positions

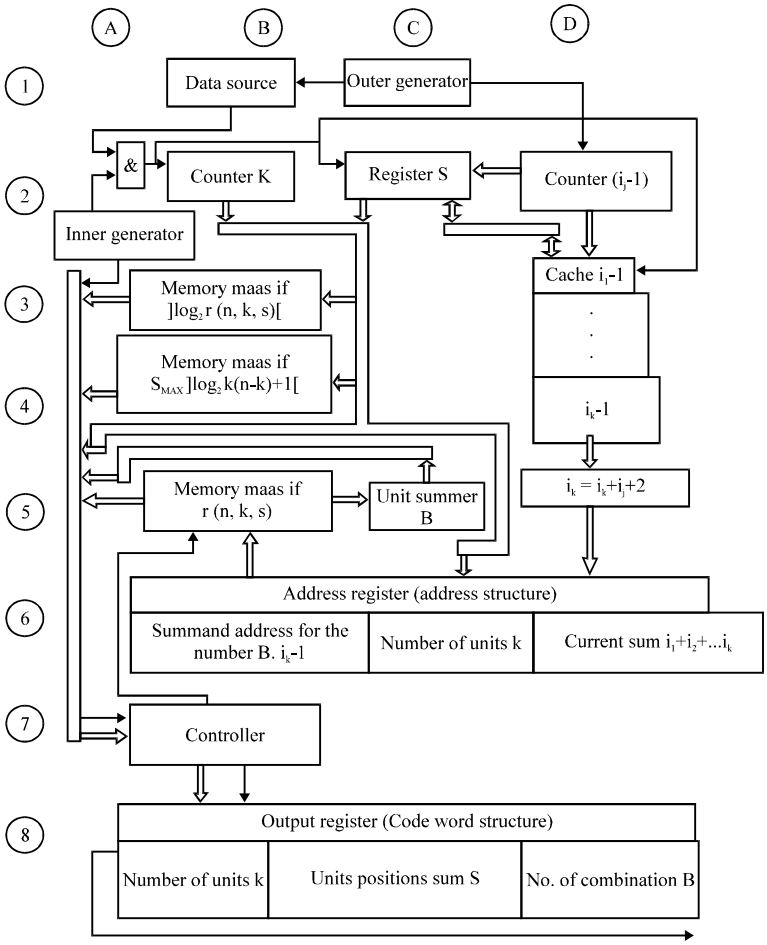

Fig. 3: Encrypting device structure scheme

In n bit block:

- A(I) is a service massif of intermediate values of b numbers
- I is a service-running index of units position number

In Fig. 3, there is a block-scheme of hypothetical integral scheme which implements Coding algorithm sh«Data source» (B1 block) and «Outer generator» (C1 block) are external in relation to microcircuit. Assuming that data is a binary sequence where each bit is received through outer generator impulse. The outgoing code binary sequence (A8 block) is sent through lockout impulse of the inner generator (A2 block).

The controller (A7 block) controls the process of receiving, coding and distribution in particular, it determines the beginning and the end of each code word parameter distribution because code words have variable length. In flash-memory blocks (A3, 4, 5) parameters are written beforehand. These parameters are needed for code word formation for n source block set length.

Faucet (A2 block) allows passage of inner generator impulse if the data source sends "one". The corresponding counters and summers (B2, C2, D2, C5 blocks) as well as a cache (D3 block) form parameters for address internal function register (A6 block) with the following address to memory massifs for extracting data needed for the code word formation.

RESULTS AND DISCUSSION

At present the leading manufacturers produce the wide nomenclature of compact flash-memory microcircuits with high functionality, internal topology and command system which help to implement different algorithms.

As an example, let us examine a microcircuit by SAMSUNG with NAND-K9LAG08U0M 16 Gbit (2048 M×8 bit) NAND Flashmemory cells. The microcircuit is placed inside TSOP-type body (box 12 mm long, 20 mm wide and 0.5 mm thick). Power voltage range is between 2.7 and 3.6 V (Datasheet K9LAG08U0M 16 Gbit (2048 M×8 bit) NAND Flash 2008).

Crystal structure is shown in Fig. 4 (Datasheet K9LAG08U0M 16 Gbit (2048M×8 bit) NAND Flash 2008). With this microcircuit, the block-scheme presented in Fig. 3 can be implemented. Command system and crystal memory volume, controller, registers needed, buffers and gates allow doing so.

It is to be recalled that during the formation of "Universal coding" (60-80'sof the XX century) the main problem for implementation of the theoretical developments was high (by then) labor intensity of coding. Nobody could imagine that 16 MB microcircuits can be created. Command system and memory volume of such microcircuit allow coding for hundreds and thousands bites long source blocks (unbelievable achievement in the past). This helps to reach theoretical maximum of binary data compression when statistics is unknown and there is a possibility of the full recovery. The scheme contains the following blocks:

- Buffers latches and decoders-buffers with "latches" and lines addresses decoder and memory massif columns
- Flash ARRAY-memory cells massif

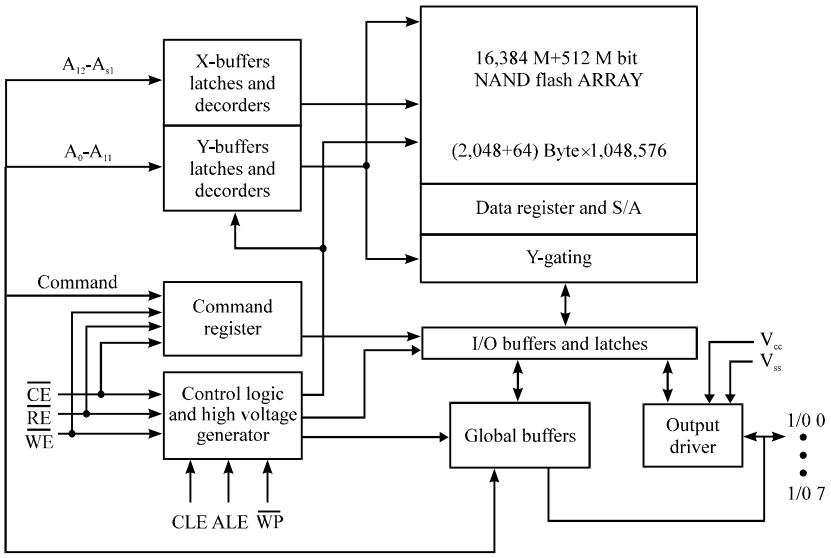

Fig. 4: K9LAG08U0M memory microcircuit block-scheme

- Data register and S/A-data register and address selector
- Cache register-"cache-register" which helps to parallelize the choice and work with cells
- Y-gating-gate which "lets through" data to memory massif and "let it go out"
- I/O buffers and latches-incoming/outgoing buffers and "latches"
- Command register-register of commands
- Control logic and high voltage generator-process controller and built-in high-voltage generator for programming
- Global buffers-"global" all-purpose buffer-registers which data (addresses, information) is determined by the controller
- Output driver-ports for external communication

To provide data transfer during reading/recording of the page, between memory cells and I/O ports of these microcircuits there are consequently connected data registers and cache registers.

Reading is done page-by-page while deleting operations are done block-by-block. It is impossible to delete separate bits. Record operation is done in 300 msec per page. Deleting is done in 2 msec per block. Data bite is read from the page in 50 nsec. I/O pins are connected to the microcircuit addressing ports. I/O of commands/addresses/data are done thorough them.

For processing record and data control there is a built-in controller on the crystal. Built-in record controller automatizes input program and deleting including impulse repetition (where necessary), inner control operations and data separation.

Microcircuit has data check system with fault correction and purding of fault data in real time. Microcircuit has 8 multi-complex address I/O. Input of commands, addresses and data is processed on the lower level on CE output, during CE fall signal, though the same I/O pins. The input information is locked on WE signal edge.

Command Lockage Enabling signals (CLE) and Address Lockage Enabling (ALE) are used to multiplex command and address accordingly though the same I/O pins.

At the same time, every microcircuit adaptation for solving "non-profile" task is hard, inner command programming demands high labor intensity, memory massif building needs to be driven towards the certain task that is why microcircuit inner resources usage and temporary work characteristics will be non-optimal.

Researchers think that optimal decision will be in creation of original microcircuit based on flash-technologies which implements Fig. 2 algorithm and Fig. 3 block-scheme.

Generally, it can be stated that method implementation helps to significantly save data storage devices memory volume and to increase channels capacity. Researchers plan to continue working on the presented method development.

CONCLUSION

The method is meant for binary data compression within the unknown statistics of message source. The usage of this method is justified when exact source binary sequence should be restored (e.g., in onboard systems of binary data collection and transfer to earth, computer archivers, electronic archives, medical devices (Kozlov *et al*., 2011), etc.), i.e., data compression at the cost of information part loss is not applicable.

REFERENCES

Alexandrovich A.Y., I.M. Yadikin and V.A. Shurigin, 2011. Binary data universal coding method. Radio Electron., 2: 94-115.

Kozlov, N.A., V.A. Shurigin, I.Yu. Zhukov, I.D. Fedorov, E.V. Ivanova and D.M. Mikhaylov, 2011. Image compression method for wireless capsule endoscopy. Specialized Mach. Commun., 6: 34-37.

Tan, G. and Y. Wang, 2009. A compression error and optimize compression algorithm for vector data. Proceedings of the International Conference on Environmental Science and Information Application Technology, July 4-5, 2009, Wuhan, pp: 522-525.

Zhang, F., L. Cheng, X. Li, Y. Sun, W. Gao and W. Zhao, 2015. Application of a real-time data compression and adapted protocol technique for WAMS. Power Syst. IEEE Trans., 30: 653-662.