PAPER Special Section on Fundamentals of Information and Communications

Hardware Implementation of the High-Dimensional Discrete Torus Knot Code

Yuuichi HAMASUNA[†a)], ***Regular Member***, **Masanori YAMAMURA**[†], **Toshio ISHIZAKA**[†], ***Nonmembers***, **Masaaki MATSUO**[†], **Masayasu HATA**[††], ***and*** **Ichi TAKUMI**[†††], ***Regular Members***

SUMMARY The hardware implementation of a proposed high dimensional discrete torus knot code was successfully realized on an ASIC chip. The code has been worked on for more than a decade since then at Aichi Prefectural University and Nagoya Institutes of Technology, both in Nagoya, Japan, [1]–[7]. The hardware operation showed the ability to correct the errors about five to ten times the burst length, compared to the conventional codes, as expected from the code configuration and theory. The result in random error correction was also excellent, especially at a severely degraded error rate range of one hundredth to one tenth, and also for high grade characteristic exceeding 10^{-6} [3]–[5]. The operation was quite stable at the worst bit error rate and realized a high speed up to 50 Mbps, since the coder-decoder configuration consisted merely of an assemblage of parity check code and hardware circuitry with no critical loop path. The hardware architecture has a unique configuration and is suitable for large scale ASIC design. The developed code can be utilized for wider applications such as mobile computing and qualified digital communications, since the code will be expected to work well in both degraded and high grade channel situations.

key words: *robust and high speed error correction, burst error, wired logic and parallel operation, ASIC*

1. Introduction

Recently, robust and high-speed correction codes have been called for in digital communications. We propose the new code that satisfies the requirements. The code has a topological structure of a discrete torus knot wound on a high-dimensional discrete cube consisting of parity check axes. The axes of the cube are n-independent party check lines. Errors produced on the code block are uniformly spread throughout the code block by the torus knot windings to achieve better corrections [3]. Error digits are detected by the n-parity check lines and then corrected by the majority logic decision with a number of the failure parity lines. Therefore, errors of any type, including random or burst and/or mixed, are randomly distributed on all parity check lines. And the code achieves a good collection ability for longer bursts that are not expected in conventional codes. Figure 1 shows a schematic diagram of the proposed code that shows data flow from input to output. A proposed code consisting of a high-dimensional discrete torus knot was successfully implemented on an ASIC through the effective development and unique hardware architecture. The code has been designed to fit recent applications such as mobile communications and reliable and high-speed digital communications. Thus, the code is tolerable even for the worst channel bit error rate on the order of one hundredth to one tenth, and also is operable irrespective of random or burst errors. Although the code framing is long and complicated due to high dimensionality, the architecture and operation of the hardware is simple and fast due to the introduction of wired logic circuitry in the implementation. This allowed us to create a smart encoding/decoding process circuit. The circuit process time is only a few clock cycles, due to the cyclic properties of the torus knot code structure.

2. Code Configuration and Characteristic

The code has an n dimensional discrete cubical structure. Each dimensional axis has a size of m, and the axis consists of m code points. Code is denoted as nDm, where n is dimension and m is size. For example, 4D5 means the code whose dimension is four and size is five. Each axis has one single redundant digit and $m-1$ data digits, satisfying the even parity. There are n such independent parity axes, forming m^n code points. The nDm code has $(m-1)^n$ data digits, totaling m^n digits including parity redundant digits, and the transmission rate is $R=(1-1/m)^n$. On these code points, the transmission order runs obliquely to form a discrete torus knot. Therefore, the errors which appear on the block are uniformly distributed on each code axis. Each code

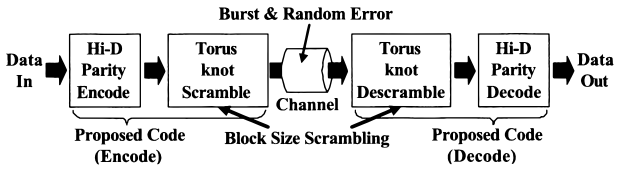


Fig. 1 Schematic diagram of proposed code.

Manuscript received August 18, 2000.
Manuscript revised November 13, 2000.

[†]The authors are with DDS Inc., Otohbashi, Nagoya-shi, 454-0012 Japan.
[††]The author is with Aichi Prefectural University, Aichiken, 480-1198 Japan.
[†††]The author is with Nagoya Institute of Technology, Gokiso, Nagoya-shi, 466-5588 Japan.
a) E-mail: hamasuna@dds.co.jp

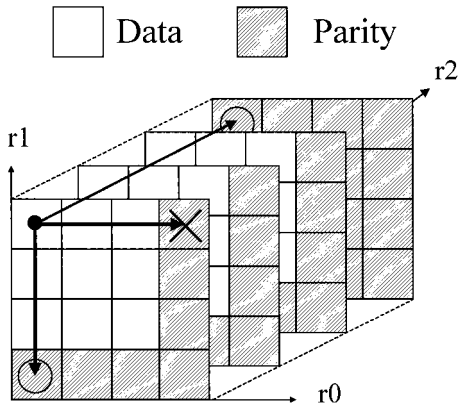


Fig. 2 Proposed code formation, an example of 3D4.

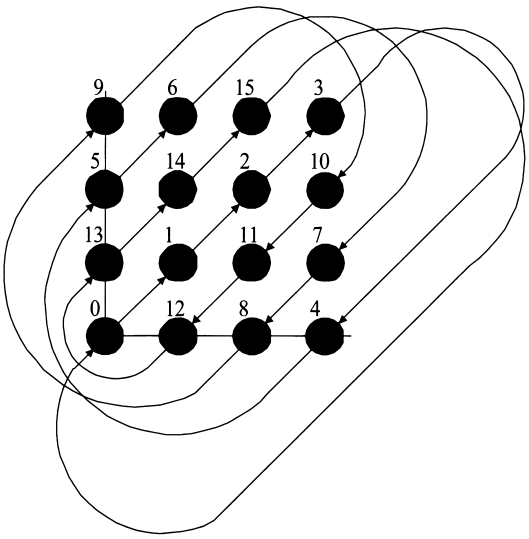


Fig. 3 Transmission order, an example of 2D4.

digit is parity checked by n intersected pairs of dimensional parity check lines. On the decoding side, each digit is checked by n parity check lines and corrected, when the number of failed parity check lines exceeds the threshold value. The decoding method was developed by Hata et al. [1].

Figure 2 shows the proposed code formation of 3D4.

Figure 3 shows the transmission order example that runs obliquely to form a discrete torus knot for two-dimensional code.

Figure 4 shows the improvements in the decoded error rate through iteration of the decoding process by varying the threshold value. The improvement in performance by iteration was introduced in [1]. Decoding ability for failure parity becomes more reliable at higher threshold setting but the code detects fewer errors, while the lower threshold picks up erroneous corrections. Therefore, in order to achieve good decoding characteristics, we should consider avoiding erroneous corrections by varying the threshold value. In Fig. 4,

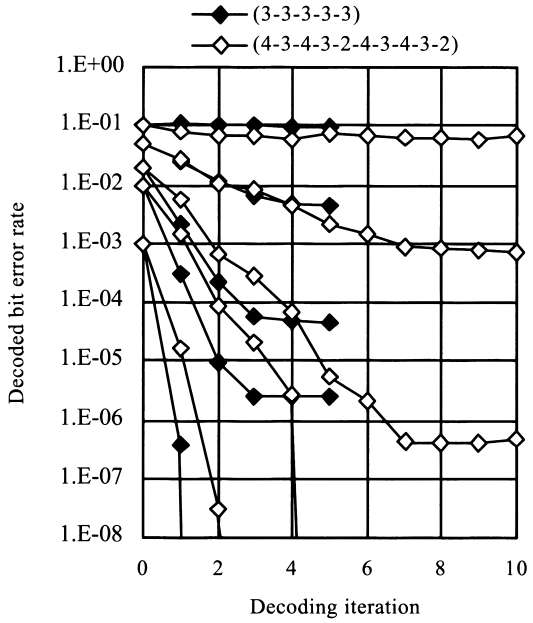


Fig. 4 Improvement in the decoded error rate for 4D5 code through iteration of the decoding process for the parameter of the initial BER at the zero iteration.

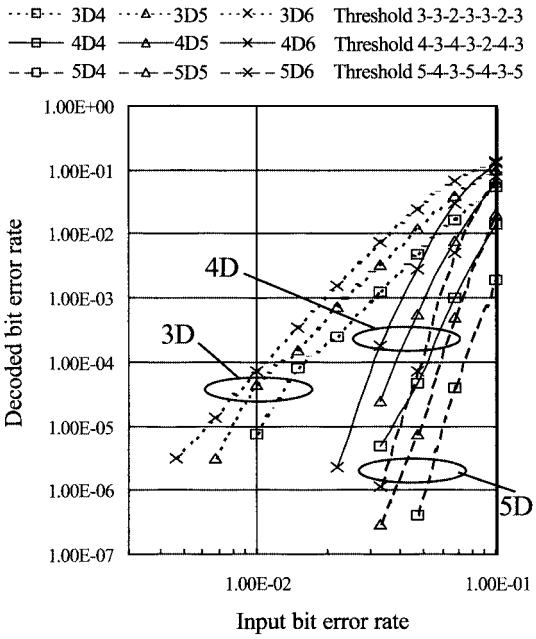


Fig. 5 Correction abilities for code parameters.

we can see that a better allotment for the threshold is (4-3-4-3-2-4-3-4-3-2) rather than (3-3-3-3-3) for the 4D5 code. The proposed code's characteristics were so excellent because the proposed code consists of an assemblage of simple parity check code and produces hardly erroneous corrections. The code does not lose the parity check function for erroneous corrections. Therefore, by adopting repeated correction, the error correction ability could be highly improved. However, there is a

Table 1 Transmission rate as a function of dimension n and size m.

nDm	Block size (bit)	Data size (bit)	Rate
3D4	64	27	0.422
3D5	125	64	0.512
3D6	216	125	0.579
4D4	256	81	0.316
4D5	625	256	0.410
4D6	1296	625	0.482
5D4	1024	243	0.237
5D5	3125	1024	0.328
5D6	7776	3125	0.402

variety of dimensions and sizes of the code, so varying the threshold to fit the channel bit error rate improves the code characteristics.

Figure 5 shows correction abilities for code parameters, and Table 1 shows the code parameters for realistic realization. The threshold value are fixed for 3D to 5D code, not optimized. Generally, larger dimension yield better performance for the same transmission rate, because errors are corrected by many parity lines.

3. Performance Simulation

In order to clarify the performance of the proposed code compared to conventional codes, we tested the code performance by software simulation for the cases of random and burst errors. The proposed code that is selected for simulation is 4D5(R=0.41, threshold=4-3-4-3-2-4-3) and 4D6(R=0.48, threshold=4-3-4-3-2-4-3-4-3-4-3-2-4-3). Selected codes for comparison were a Reed-Solomom code and Convolutional code with Viterbi decoding and Turbo code.

The Reed-Solomon code's rate is 0.48, and the length of codeword is 15 and the length of data is 7 in extension field of 2^3. The encoding is in systematic form, and the decoding is via the Berlekamp iterative algorithm. The software of the Reed-Solomon code owes to Simon Rockliff [8].

The Convolutional code is a half rate with constraint length seven, which was mounted with Viterbi decoding. The two generator polynomials for the NASA Standard $K = 7$ code are used in the Viterbi decoding. Since these polynomials are known to be optimal for this constraint length there is not much point in changing them. The software of the Convolutional code is owing to Phil Karn [9].

The turbo code with a half rate is mounted with the Log-Map algorithm. The numbers of iteration are 5 and 18, it's frame size are 624 bit and 65,536 bit. It simulates parallel concatenated convolutional codes. Two component rate 1/2 RSC (Recursive Systematic Convolutional) component encoders are assumed. The software of the turbo code owes to Yufei Wu [10].

The test data file was prepared using a visual moving picture of a toy train as in Fig. 6. The digital data was generated by MPEG-2 band compression technology and the file size was 3,724,280 Bytes, or approximately 30 Mbit.

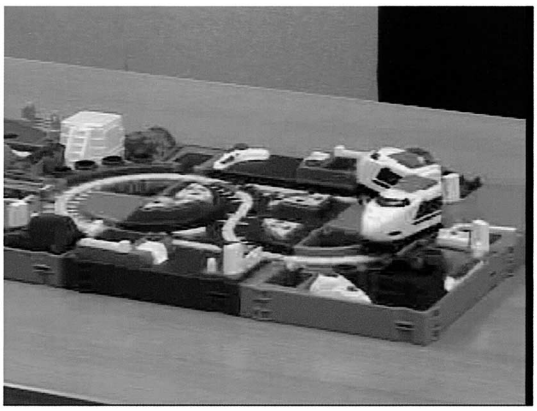


Fig. 6 Mpeg-2 file data for the performance simulation.

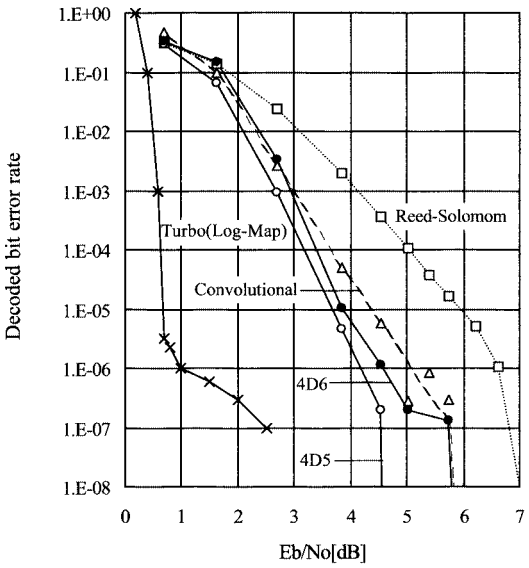


Fig. 7 BER improvements for random error.

3.1 BER Improvements for Random Error

Figure 7 shows the decoded error rate for random errors. The simulation was evaluated for the practical 30 Mbit MPEG2 file data, and 10^5 code blocks were tested. The figure estimates the performances by Eb/No. The performance is superior to that of the conventional codes. On the other hand, as compared with the excellent data of the turbo code, the random error correction ability measured at the BER of 10^{-5} degrades by approximately 3.0 dB. The turbo code's parameters are LOG-MAP decoding algorithm with 18 iteration, block length of 65,536, code generator g=(037,021) and with half rate. Future potentiality of the code is remained for coding modulation gain.

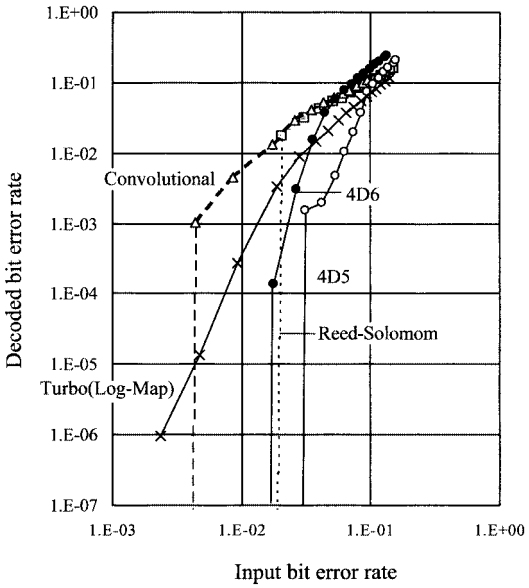
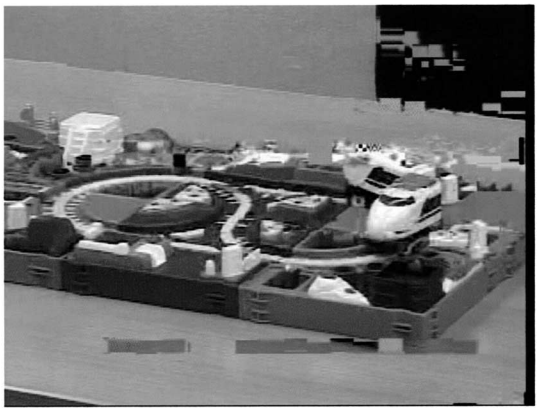


Fig. 8 BER improvements for burst error.

3.2 BER Improvements for Burst Error

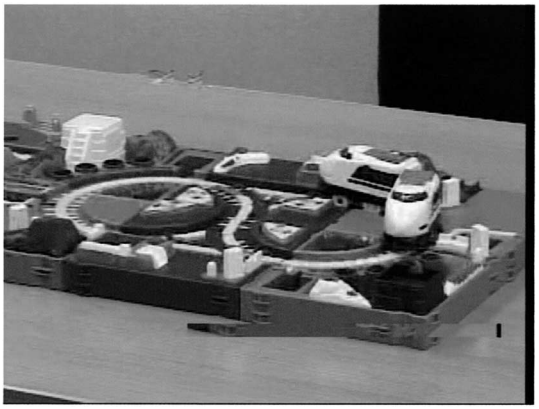
Figure 8 shows a comparison for burst errors. The burst error was introduced with all marked digits of length from 4 to 240 digits for each 800 digits of 30 Mbyte MPEG2 file data. To estimate the burst error correction ability for various burst error positions in the block, the length of the error block added to the code block was selected as the 800 bits, different from the 625 bits of the code The hundred thousand code blocks are tested in the simulation. The remained error bits are measured and the ratio to the code block is obtained as the decoded BER for burst error of each burst error length of 4,8,16,32 and to 240 bits at an interval of 16 bits. For burst error, the performances are compared as a function of input error rate. In this paper, the burst error is often expressed not by the length but by the counted bit error rate where the error bits are divided by the total number of digits. For the comparison, we make use of half rate turbo-code of almost the same block length of 624 bits with that of the proposed code and of the code generator g=(111;101).

The proposed code shows remarkable properties in correcting burst errors compared to the conventional codes and turbo code. We can see the proposed code's remarkable abilities to handle burst errors up to three percent of the input bit error rate. Convolutional code and Turbo code, on the other hand, deteriorate themselves at an input error rate of less than one percent. Although the Reed-Solomon code is much stronger than the convolutional code in the faint burst error ranges, it deteriorates its ability more rapidly for the input rate, exceeding 0.015.



(1) Decoded by Reed-Solomon

(2) Decoded by Viterbi



(3) Decoded by 4D5 torus code

Fig. 9 Decoded image for mixed random and burst error. Total equivalent BER is 0.016.

3.3 Mixed Random and Burst Error Simulation

Figure 9 shows the visual estimation of the code correction ability for a severe error situation. The test data was the toy train visual data processed by the MPEG-2 band compression technique, and it was embedded with both random and burst error digits. Random errors had an error rate of 0.0125, and burst errors had a

marked interference of length six, which was repeatedly introduced every 800 bits. The total bit error rate calculated by the number of errors was 0.016. The band compression data is sensitive to error and the worst case was the Convolutional code with Viterbi decoding, followed by the Reed-Solomon. The proposed code case was shown to be tolerable for some practical use.

4. Hardware Architecture

Hardware implementation of the code has been aimed at since 1998 [2], but we were unable to develop a hi-dimension, hi-speed code circuit due to the complexity of the circuit. This time, we solved this problem with the new system for implementation.

The high-dimensional torus knot code consists of a simple parity check code. Therefore, we can find out the simple relation between parity and data digits. A large part of the encoding and decoding processes are substituted by adopting wired connections between memory cells, and we could acquire speeding up the operation of the code. We utilized the hardware description language VHDL, which translates software programs into hardware configurations. Installation of hardware circuitry through software language is especially suitable for circuitry that consists of repetition or regularity. We developed a program to generate a source program for the VHDL automatically, depending on the code dimension and size. By making use of the program, it is possible to generate the encoder and decoder for three to five dimensional codes.

4.1 Encoding Configuration

Figure 10 shows the encoding configuration. The input data are first stored to the input shift register. Upon completion of storage of one block, contents are transferred in parallel to the parity calculation circuits, and the value of parity is stored in the corresponding cell of the output torus-connected shift register. The value of data is transferred to the corresponding cell of the output torus-connected shift register as well. In the torus-connected shift register, the wire connection between cells determines the order of transmission of the cell contents and eventually forms a torus knot winding.

Figure 11 shows the encoding time chart. Most of the times are the read-out times of the output shift register, and the process time in the parity calculation takes only a few clock cycles, including circuit delay.

4.2 Decoding Configuration

Figure 12 shows the decoding circuit. The received data are fed to the memory cells of the torus connected shift register through the torus loop wired connection as the order of the encoder output shift register. The register

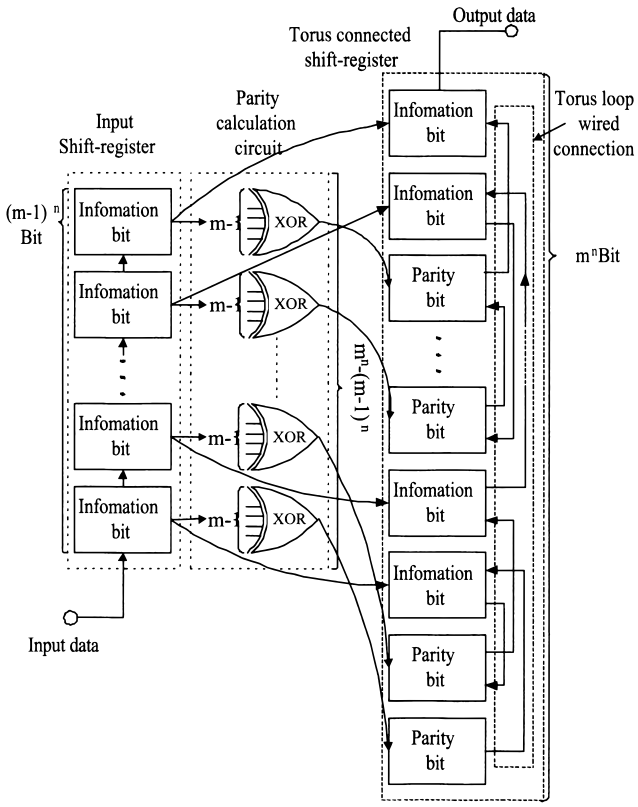


Fig. 10 Encoder configuration.

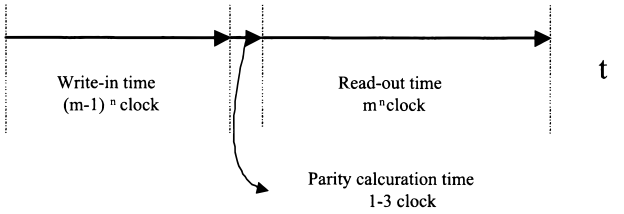


Fig. 11 Encoding time chart.

contents are transferred in parallel to the parity calculation circuit and also to the majority logic circuit to decide the digit. The majority logic circuit is a principal part of decoding, and each digit is corrected by majority decision of the n independent parity checks, whether or not the value exceeds the given threshold. To improve the decoding characteristics, the decision in the majority logic circuit is repeated several times by varying the threshold value, and the digit value determined by the logic circuit is fed back to the input side register. After completion of decoding, only the data digit is transferred to the output shift register.

Figure 13 shows the majority logic circuit containing repeatable decoding by varying the decision threshold value.

Figure 14 shows the decoding time chart. The time required for decoding is quite small, considering the multiple decoding. Just 5-35 clock cycles is sufficient

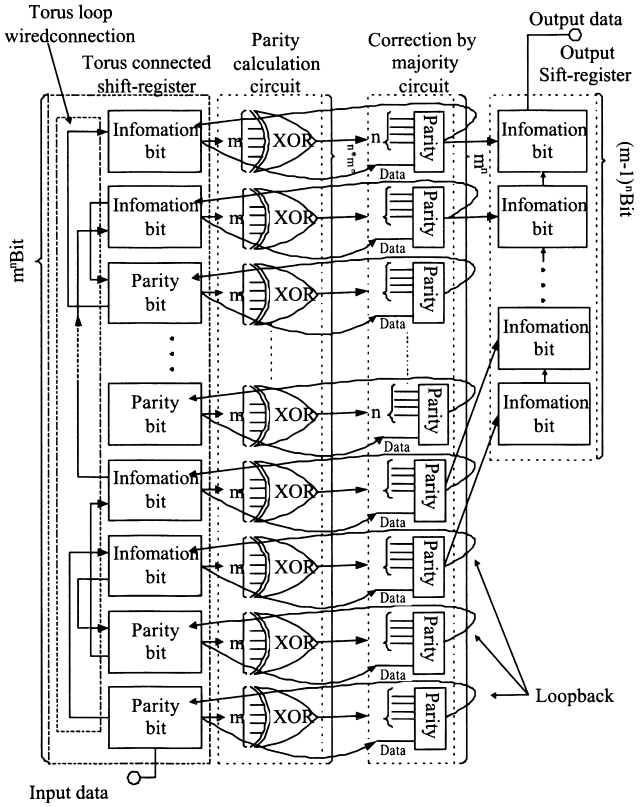


Fig. 12 Decoder configuration.

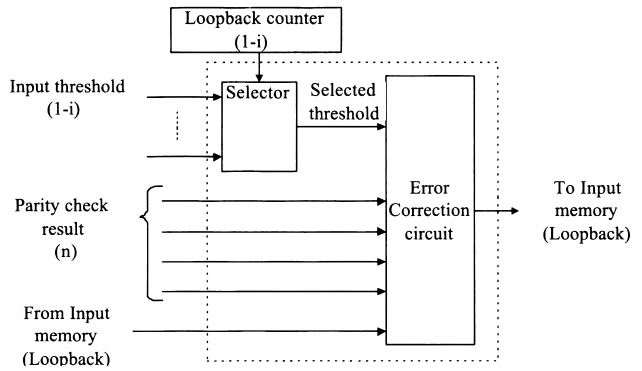


Fig. 13 Majority correction circuit.

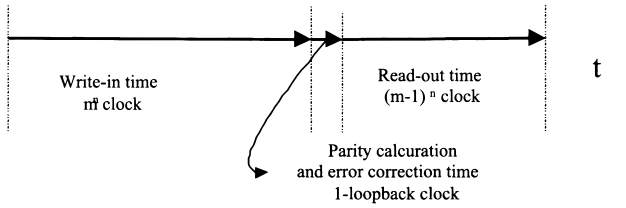


Fig. 14 Decoding time chart.

to decode, which is negligible as the read-out time.

The special algorisms of the code were simplified by the wired logic and parallel operation basis of the processes. The wired logic is effectively supported by the simplicity of the circuit configuration. Therefore, the dimension and size of the code is not an essential limitation on the operation speed. In short, the speed does not depend on the code dimension and size.

Table 2 Outline of developed-chip for 4-dimension code.

Chip-spec	
Circuit Size	50 kGate NAND
Process	0.6 μm
Chip Size	31.2×31.2 mm
Package	QFP120P
Clock	Speed 50 MHz
Through-put	6.25 Gbps
Input-Output	50 Mbps (Serial), 50×8 bps (Parallel)

Code-Performance	
Code	4-Dimension Size 5(4D5)
Block Size	625 bit
Data Size	256 bit
Rate	0.41
Correctable Mean BER	0.021 (Burst and Random)

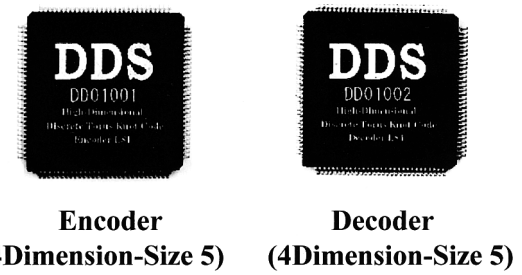


Fig. 15 Developed 4D5 code chip.

5. Outline of Developed-Chip

The outline of the LSI chip of the 4 dimensional size 5 code is shown in Table 2. The LSI chip consists of 50 k gate of 0.6 μm, and has an input-output speed of 50 Mbps and a decoding throughput of 6.25 Gbps. For the LSI chip, it was unable to implement the full function of the 4D5 code on the 50 k gates NAND for lack of small amount of gates. And so, the function was divided to five processes. Therefore, the decoding process took five clocks, and then the throughput came to 6.25 Gbps. The next version will attain 5 times speed up by adopting a larger scale LSI chip. The code perfectly corrected the burst errors of length 32 bit with the mean block BER of 2.1×10^{-2}, regardless of random or burst ones. Figure 15 is a developed 4D5 code chip. The LSI chips have been successfully applied to a trial radio transmission equipment of image of 10 Mbps.

We tested the developed code chip by the test board. The test board encodes the input data by making up use of the LSI chip. Figure 16 is a Test Board that is made for the developed chip's test. Figure 17 shows the simple structure of test system. The en-

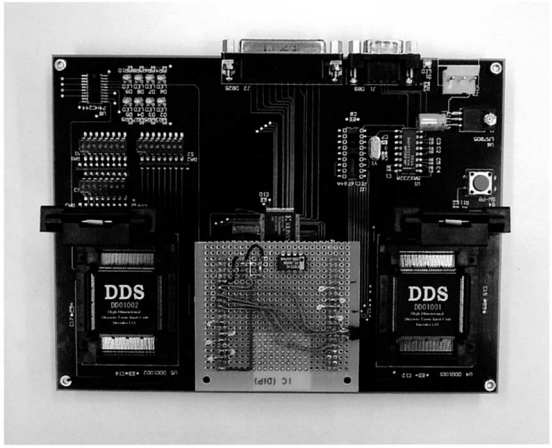


Fig. 16 Test board. (4D5 encoder and decoder)

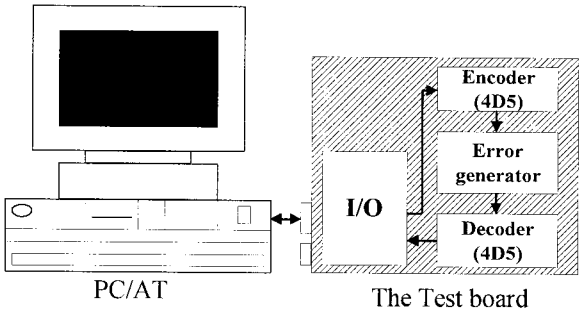


Fig. 17 Test system for the developed code chip.

coded data are degraded by adding the random error produced by m sequence generator of 17 stage (128 k length). The decoding performance by the chip showed almost the same value with the software simulation results with perfect correction to about 2 percent BER.

6. Conclusion

We proved that the hardware implementation of a trous knot code was effectively realized by using a unique circuit configuration, which maximizes the cyclic and symmetrical properties of the code. Furthermore, through performance simulations, the code was proven to have robust decoding characteristics in such severe channel bit error rates as one hundredth to one tenth, as well as for longer burst errors. The high dimensional torus knot code is expected to find wide applications in the field of digital communications in the near future. The work was supported by the '98 NEDO project of the Ministry of International Trade and Industry of Japan.

References

[1] M. Hata, E. Yamaguchi, I. Takumi, Y. Hamasuna, and K. Miyoshino, "Decoding algorithm for the high-dimensional discrete torus knot code," Code in Non-Hamming Spaces p.59, ISIT 2000, Sorrento, Italy.

[2] M. Hata, E. Yamaguchi, A. Ando, and I. Takumi, "High-dimensional discrete torus knot code," Block code p.18, ISIT 1998, Cambridge, MA, USA.

[3] S. Kuroda, E. Yamaguchi, I. Takumi, and M. Hata, "A Geometrical decoding algorithm and correcting limit of high-dimensional hyper-cubic ring code—Correcting ability in BER of $10^{-2} \sim 10^{-1}$," IEICE Trans., vol.J80-A, no.12, pp.2145–2154, Dec. 1997.

[4] M. Hata and I. Takumi, "High-dimensional symmetric compacted code, —Error-correcting of high bit error rate of $10^{-2} \sim 10^{-1}$," ISIT93, p.297, Jan. 1993.

[5] K. Hashimoto and M. Hata, "High-dimensional symmetry parity check code capable of correcting $10^{-2} \sim 10^{-1}$ random errors," IEICE Trans., vol.J75-A, no.8, pp.1257–1266, Aug. 1992.

[6] T. Noda and M. Hata, "Ring code-code symmetry and uncorrectable errors," IEICE Trans., vol.J73-A, no.2, pp.243–252, Feb. 1990.

[7] M. Hata and I. Takumi, "A remainder theorem of arbitrary integer module M with cyclic shift carry and its application to quasi-cyclic code," ISIT88, p.174, June 1988.

[8] Copyright Simon Rockliff, University of Adelaide, 1991. http://imailab-www.iis.u-tokyo.ac.jp/Members/robert-e.html

[9] Copyright Phil Karn, KA9Q, 1995. http://imailab-www.iis.u-tokyo.ac.jp/Members/robert-e.html

[10] Copyright Yufei Wu, MPRG lab, Virginia Tech,1998, http://www.ee.vt.edu/yufei/turbo.html

Yuuichi Hamasuna received the M.S. (1993) and B.E. (1991) in Electronic Engineering, Saga University, Saga, Japan. He joined DDS Inc. in 1995 and works on R&D of Error Correction Technology.

Masanori Yamamura received the B.S. in Economics (1993), Saga University, Saga, Japan. He joined DDS in 1995 and works on developing Error Correction Technology and Image Recognition Technology.

Toshio Ishizaka received the M.S. (1999) and B.E. (1997) in Computer Science, Kyushu University, Fukuoka, Japan. He joined DDS Inc. in 1999 and works on developing Image Recognition Technology.

Masaaki Matsuo is a doctoral candidate at Saga University, Saga, Japan. He received the M.S. (1993) and B.E. (1991) in Electronic Engineering, Saga University, Saga, Japan. He joined DDS Inc. in 1995 and works on developing a variety of software.

Masayasu Hata is professor of Engineering, Aichi Prefectural University, Aichi, Japan. He is a doctor of Engineering, Tokyo Institute of Technology, Tokyo, Japan. Graduated from Nagoya Institute of Technology in Electronic Engineering. Joined Oki Electric Co. and worked on R&D of digital communication systems, application of electronic circuits, and millimeter wave communication equipments. Left Oki Electric Co. in 1985 and became Professor in Department of Artificial Intelligence and Computer Science at Nagoya Institute of Technology, Nagoya, Japan. Currently works on digital processing and information communications.

Ichi Takumi is an associate Professor of Engineering, Nagoya Institute of Technology, Nagoya, Japan M.S. (1984) and B.E. (1982) in Electronic Engineering, Nagoya Institute of Technology. Joined Oki Electric Co. after graduation. Currently works on digital signal processing and digital communications. Member of the Society of Instrument and Control Engineers of Japan.