

UDC 004.7

**Oleksandr Honcharenko,
Artem Volokyta, Heorhii Loutskii**

**FAULT-TOLERANT TOPOLOGIES SYNTHESIS
BASED ON EXCESS CODE USING THE LATIN SQUARE**

**Олександр Гончаренко,
Артем Волокита, Георгій Луцький**

**СИНТЕЗ ВІДМОВОСТІЙКИХ ТОПОЛОГІЙ НА ОСНОВІ
НАДЛИШКОВОГО КОДУВАННЯ
ЗА ДОПОМОГОЮ ЛАТИНСЬКОГО КВАДРАТУ**

The article discusses the method of synthesis of fault-tolerant topologies using a Latin square and excess encoding of nodes numbers. Ways to fill the square were considered, several topologies were synthesized and their characteristics were analyzed. The usages of redundancy have been analyzed.

Key words: fault tolerance, excess code, Latin square

Fig.: 5. Tabl.: 5. Bibl.: 6.

У статті розглядається метод синтезу відмовостійких топологій за допомогою латинського квадрату з використанням надлишкового кодування номерів вершин. Було розглянуто способи заповнення квадрату, синтезовано кілька топологій і виконано аналіз характеристик. Проаналізовано можливості використання надлишковості.

Ключові слова: відмовостійкість, надлишковий код, латинський квадрат

Рис.: 5. Табл.: 5. Бібл.: 6.

Urgency of the research. In modern world the distributed computing is an important branch of the development of computer technology. Improving of their fault-tolerance is a one of most important tasks. One of perspective methods of its solution is a method of fault-tolerant topologies synthesis, that allow hardwarely provide a high level of systems fault-tolerance. In article discusses the method of fault-tolerant topologies synthesis with excess code 0/1/-1 and Latin square.

Target setting. A topological structure is an important part of distributed computing system. A lot of parameters depends from it, including fault-tolerance. There are some methods of fault-tolerant topologies synthesis, one of these is a using of an excess code 0/1/-1 in nodes numbers encoding. This article proposes a method of fault-tolerant topologies synthesis with Latin square and excess encoding.

Actual scientific researches and issues analysis. Now a topologies synthesis with Latin square was good considered. Also method of topologies synthesis with excess encoding was proposed [1], the synthesis of fault-tolerant versions of hypercube and quasi-quantum topologies was completed. In previous publications was performed an analysis of main advantages and disadvantages of these topologies, proposed methods of using redundancy to improving of fault-tolerance.

Uninvestigated parts of general matters defining. In last time only a synthesis of fault-tolerant topologies based on codes transformations was consider, a using of Latin square and other methods of synthesis without explicit nodes codes transformations wasn't considered.

The research objective. The purpose of the research is a consideration and analysis of possibilities of using Latin square for topologies synthesis with excess encoding of node's numbers, creating examples of these topologies and its analysis.

The statement of basic materials. The problem of synthesis with the square in its usual form is as follows: there are Latin square. In first column – numbers of nodes, for which the definition of neighbors is carried out. The order in which the nodes in the column are listed can be any, it is important that the code of each node is encountered only once. Next, the sequence written in the first column is cyclically shifted up and written to the second column. This doing repeats for next columns analogically until the square been filled. All cols except first are the possible neighbors of nodes, written in first column. After the square creating from possible neighbors in any way, several columns may be selected and a topology is constructed on them.

Basic definitions. The excess code 0/1/-1 has the same parameters as the usual binary, but contains an additional digit -1 denoted by the letter T [1]. The main feature of this code is that one and the same number has several possible views in this code. That's what makes it excess.

Topologies that use this code as a basis have high fault tolerance. This is ensured, first, by the presence of nodes with the same number, and secondly, using routing trees to bypass failures in the system. But, as a rule, the disadvantage of such topologies is too high a power or diameter. In synthesis based on transformations, this is due to the fact that the transformation must take into account an additional number -1, which increases the number of results. For example, on Exchange transformation it leads to the appearance of several results. Exchange with the lowest digit of code 010 gets not only 011, but also 01T. In the framework of transformations-based synthesis paradigm nothing can be done. An alternative to this is a synthesis without transformations.

An example of topology synthesis using Latin square method. In order to construct a Latin square, it is necessary and sufficient to determine the sequence of node's codes entries in the first column. For example, tables 1 and 2 illustrates the Latin squares, constructed for natural sequence and for Gray's code.

To perform the synthesis of the topology, you must select columns in square. For example, select cols 1 and 3. For first square this means, that neighbors of node

000 are nodes 001 and 011, neighbors of 001 – 010 and 100, etc. For second square neighbors of node 000 are nodes 001 and 010, for 001 – 011 and 110, etc.

Table 1

Latin square for 3-bits binary code, written in natural sequence

№	Node	Possible neighbors						
		1	2	3	4	5	6	7
0	000	001	010	011	100	101	110	111
1	001	010	011	100	101	110	111	000
2	010	011	100	101	110	111	000	001
3	011	100	101	110	111	000	001	010
4	100	101	110	111	000	001	010	011
5	101	110	111	000	001	010	011	100
6	110	111	000	001	010	011	100	101
7	111	000	001	010	011	100	101	110

Table 2

Latin square for 3-bits binary code, written in Gray's code sequence

№	Node	Possible neighbors						
		1	2	3	4	5	6	7
0	000	001	011	010	110	111	101	100
1	001	011	010	110	111	101	100	000
2	011	010	110	111	101	100	000	001
3	010	110	111	101	100	000	001	011
4	110	111	101	100	000	001	011	010
5	111	101	100	000	001	011	010	110
6	101	100	000	001	011	010	110	111
7	100	000	001	011	010	110	111	101

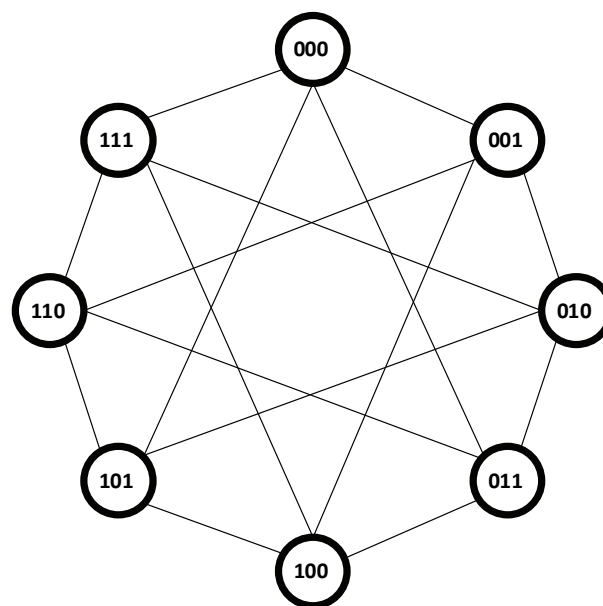


Fig 1. Topology, built with first Latin square with selected cols 1 and 3

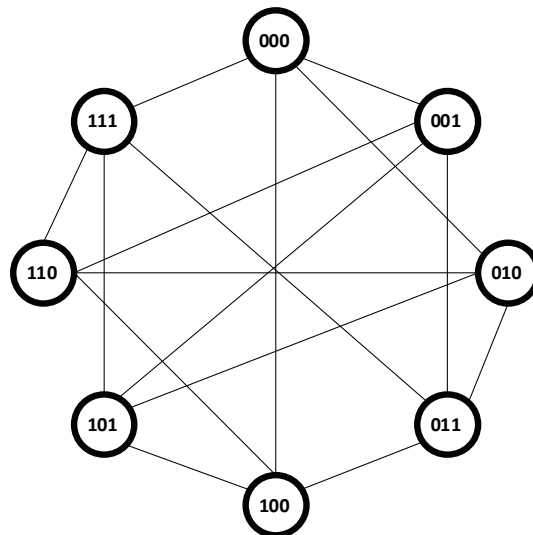


Fig 2. Topology, built with second Latin square with selected cols 1 and 3

The most interesting thing in this synthesis is that both topologies have power 4. Theoretically, in some cases, the power of the synthesized topology can be less than 4, but not more: in 2 columns opposite each node 2 neighbors and the node itself in each of the two columns opposite some other occurs only once.

Also the interesting thing is that obtained topology depends not only from selected columns, but also from sequence in first column.

Also, there is an interesting property in second topology: 3 links of every node analogical to links in hypercube, and 4th link can be received by the way of all code's bits' inversion. It means that routing methods, similar for routing in hypercube, can be used for this topology. For comparison, in table 3 the second topology's system of links described in terms of functional dependencies between node's codes.

Table 3

System of links in second topology in terms of codes transformation

<i>Node</i>	<i>Exchange</i>			
	<i>1st bit</i>	<i>2nd bit</i>	<i>3rd bit</i>	<i>All bits</i>
000	100	010	001	111
001	101	011	000	110
010	110	000	011	101
011	111	001	010	100
100	000	110	101	011
101	001	111	100	010
110	010	100	111	001
111	011	101	110	000

Topology synthesis for excess code. In essence, synthesis occurs similarly, but there are differences. There is no natural sequence of excess codes, because some numbers in it repeat with different code, as result, all codes can't be located so that the

value of each next be more than 1. It isn't critical for synthesis, but it can be important for filling the square, because, as it was being described above, properties of topology depend from square's filling. For synthesis use one of possible pseudo-natural sequence for 2-bits excess code. Table 4 illustrates Latin square for this sequence.

Table 4

Latin square for pseudo-natural sequence of the excess codes

N_0	node	Neighbors							
		1	2	3	4	5	6	7	8
-3	TT	T0	T1	0T	00	01	1T	10	11
-2	T0	T1	0T	00	01	1T	10	11	TT
-1	T1	0T	00	01	1T	10	11	TT	T0
-1	0T	00	01	1T	10	11	TT	T0	T1
0	00	01	1T	10	11	TT	T0	T1	0T
1	01	1T	10	11	TT	T0	T1	0T	00
1	1T	10	11	TT	T0	T1	0T	00	01
2	10	11	TT	T0	T1	0T	00	01	1T
3	11	TT	T0	T1	0T	00	01	1T	10

For example, also select columns 1 ra 3. The blue dotted line highlights the connections that pass between nodes with identical numbers.

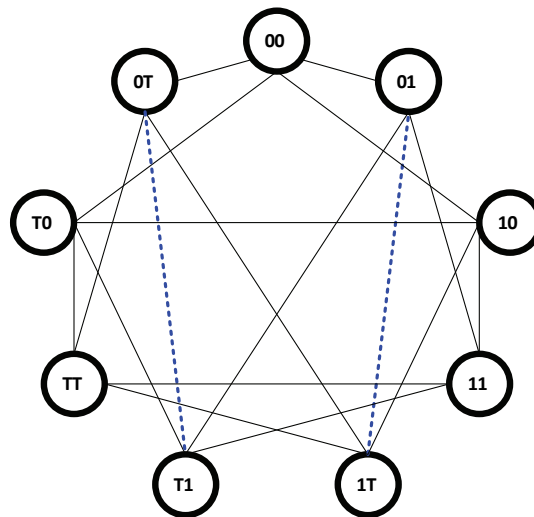


Fig 3. Topology, built with Latin square with a pseudo-natural sequence of excess codes

There is interesting property in this topology: all connections between nodes except 0T – T1, 01 – 1T, 11 – TT complete between nodes with codes, that differs by one bit. In allows in some cases use the routing algorithms, that similar to routing in hypercube.

Using of redundancy. In terms of using the redundancy obtained topology interesting is that has connections between nodes with same number. The reason for

these connections is to select column 1 for the synthesis of the topology: since in sequence these nodes are adjacent, then they are, in this way, adjacent in the topology. The presence of these links allows clustering.

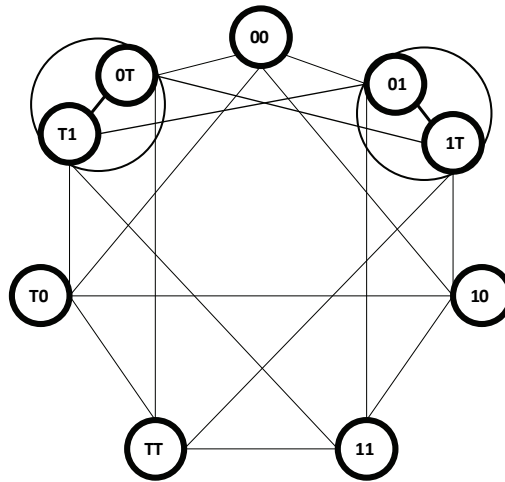


Fig 4. Topology clustering

As you can see, clusters have not only links inside, but also links between. So, even on case of fail one node of cluster this link not be broken. Moreover, if for nodes in cluster somehow provide possibility to intercept packets, that appointed to other node in cluster, every node can be substitute other in case of failure and to reroute packets, that were sent through failed node

Routing in redundant topology. As in the case of hypercube and de Bruyn topology, routing trees can be used in this topology too. It allows use all advantages of this routing method: bypass the failures and avoid locks.

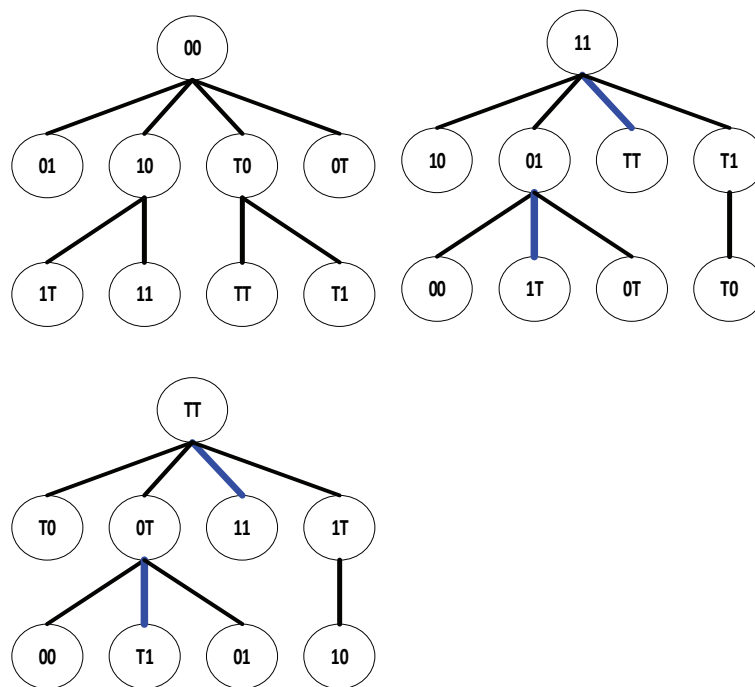


Fig 5. Routing trees for obtained topology

Comparison of the characteristics of the obtained topology. Was completed the comparison between obtained topology with same topology, based on usual binary code and with classic hypercube. In table 5 the results of this comparison are presented.

Table 5

Topologies comparison

<i>Topology</i>	<i>Classic Hypercube</i>			<i>Latin square with natural codes sequence</i>			<i>Redundant Latin square</i>					
Selected cols	-			1, 3			1, 3 (example above)			1, N/3		
Count of bits (N)	3	5	8	3	5	8	2	3	5	2	3	5
Count of nodes	8	32	256	8	32	256	9	27	243	9	27	243
Diameter	3	5	8	2	6	44	2	5	40	2	5	40
Power	3	5	8	4	4	4	4	4	4	4	4	4
Diameter *Power	9	25	64	8	24	176	8	20	160	8	20	160
Count of edges	12	80	1024	16	64	512	18	56	486	18	56	486
Topology	Redundant Latin square											
Selected cols	1, 3, 6, ..., N/3						1, 3, 9, ..., 3 ^{N-1}					
Count of bits (N)	2	3	5	2	3	5	2	3	5	2	3	5
Count of nodes	9	27	243	9	27	243	9	27	243	9	27	243
Diameter	2	3	5	2	3	5	2	3	5	2	3	5
Power	4	8	54	4	8	54	4	6	10	4	6	10
Diameter *Power	8	24	270	8	24	270	8	18	50	8	18	50
Count of edges	18	108	6561	18	108	6561	18	81	1215	18	81	1215

Conclusions. In article proposed method of fault-tolerant topologies synthesis, based on using excess code, with Latin square. Considered the clustering as method of fault-tolerance improving, received the routing trees, that can be used in this topology. Analyzed the characteristics of redundant topologies, that can be received by using Latin square, their comparison with each other was carried out.

A main advantage of method is that in the process of synthesis it is possible to directly determine the desired power of topology. Also, sequence in basis of Latin square allows to delegate some features to topology, that also can be used. Defined, that in these topologies is possible to use routing, based on trees, that allows bypass failures and provides a high level of fault-tolerance. Another advantage of the method is that all the nodes of the resulting topology often have the same power.

But there are disadvantages too. First, obtained topologies not always can be described through transformations with node's codes. This limits the ability to use some routing methods. At second, there are difficulty with defining diameter of topology before synthesis.

There are some ways to improve the method. Firstly, a features of topology depends by sequence in first column of Latin square. Theoretically, it possible to give some features to topology through choosing of sequence. The selected columns are important too. As the comparison showed, changing the numbers and the count of selected columns can be manipulated by power and diameter.

References

1. Goncharenko Olexandr, Pavlo Rehida, Artem Volokyta, Heorhii Loutskii, and Vu Duc Thinh: Routing Method Based on the Excess Code for Fault Tolerant Clusters with InfiniBand. *Advances in Intelligent Systems and Computing*, vol. 938, pp. 335-345. Springer, Heidelberg (2019)
2. Washington N., Perros H.: Performance Analysis of Traffic-Groomed Optical Networks Employing Alternate Routing Techniques. *Lecture Notes in Computer Science*, vol. 4516, pp. 1048-1059. Springer, Berlin, Heidelberg (2007).
3. Hu, Z., Mukhin, V., Kornaga, Y., Volokyta, A., & Herasymenko, O. The scheduler for distributed computer systems based on the network centric approach to resources control. In: *Proceedings of the 2017 IEEE 9th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2017*, pp. 518-523(2017).
4. Emanouilidis, E and Bell, R. Latin squares and their inverses. *Math. Gaz.*, vol 88(511), pp. 127–128 (2004)
5. Richard J.Cole, Bruce M.Maggs, Ramesh K.Sitaraman, On the Benefit of Supporting Virtual Channels in Wormhole Routers, *Journal of Computer and System Sciences*, vol. 62(1), pp 152-177 (2001)
6. Ian M. Wanless. Cycle Switches in Latin Squares, Graphs and Combinatorics, vol. 20(4), pp 545-570 (2004).

Autors

Olexandr Goncharenko – student, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute” (Solomenskiy district, ave. Pobedy, 37, 03056, Kyiv, Ukraine).

E-mail: alexandr.ik97@ukr.net

Гончаренко Олександр Олексійович – студент, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» (Солом'янський район, пр-т Перемоги, 37, м. Київ, 03056, Україна).

Волокита Артем Миколайович – доцент, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Volokyta Artem – associate professor, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: artem.volokita@kpi.ua

Луцький Георгій Михайлович – професор, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Loutskii Heorhii – professor, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

РОЗШИРЕНА АНОТАЦІЯ

**Олександр Гончаренко,
Артем Волокита, Георгій Луцький**

СИНТЕЗ ВІДМОВОСТІЙКИХ ТОПОЛОГІЙ НА ОСНОВІ НАДЛИШКОВОГО КОДУВАННЯ ЗА ДОПОМОГОЮ ЛАТИНСЬКОГО КВАДРАТУ

Актуальність теми дослідження. В сучасному світі розподілені обчислення є важливою галуззю розвитку обчислювальної техніки. Підвищення їх відмовостійкості - одна із найважливіших задач. Доволі перспективним методом її вирішення є синтез відмовостійкої топології для системи. В даній статті розглядається метод синтезу відмовостійких топологій з використанням надлишкового коду 0/1/-1 та латинського квадрату.

Постановка проблеми. Топологічна структура є важливою частиною розподіленої обчислювальної системи. Від неї залежить маса параметрів, включаючи відмовостійкість. Є кілька способів синтезу відмовостійких топологій, одним із яких є використання надлишкового кодування 0/1/-1 в нумерації вершин. В даній статті пропонується метод синтезу відмовостійких топологій за допомогою латинського квадрату і надлишкового кодування.

Аналіз останніх досліджень і публікацій. На даний момент загальновідомим методом є синтез топологій за допомогою латинського

квадрату. Також розроблено метод синтезу топологій з використанням надлишкового кодування [1], виконано синтез відмовостійких версій гіперкуба та квазі-квантової топології. В попередніх публікаціях було виконано аналіз основних переваг та недоліків таких топологій, запропоновано методи використання надлишковості для збільшення відмовостійкості.

Виділення недосліджених частин загальної проблеми. До цього часу розглядався лише синтез відмовостійких топологій на основі перетворень, не розглянутим залишається використання латинського квадрату та інших методів створення топологій, які не використовують явних перетворень коду вершин для отримання сусідів кожної вершини.

Постановка завдання. Завданням є розгляд та аналіз можливостей використання латинського квадрату для синтезу топологій з надлишковим кодуванням номерів вершин, синтез таких топологій та їх аналіз.

Викладення основного матеріалу. Задача синтезу за допомогою квадрату у звичайному вигляді полягає в наступному: є латинський квадрат. В першому стовбці – номери вершин, для яких проводиться визначення сусідів. Порядок, в якому перелічуються вершини в стовбці, може бути будь-яким, важливо, щоб код кожної вершини зустрічався в ньому лише раз. Наступний стовбець визначається так: послідовність, що записана в першому стовбці, циклічно зсувається вгору і записується в другий стовбець. Для наступних стовбців аналогічно, і так до тих пір поки квадрат не буде заповнено. Всі стовбці, крім першого, - можливі сусіди вершин, записаних в першому стовбці. Після формування латинського квадрату із можливих сусідів будь-яким чином обирається кілька стовбців і по ним будується топологія. Основна відмінність запропонованого методу – в використанні надлишкового коду для формування квадрату.

Висновки. Виділено основні особливості запропонованого методу. Виконано синтез топології для прикладу, показано застосування надлишковості для підвищення відмовостійкості. Проведено аналіз характеристик топологій, що можуть бути синтезовані з використанням запропонованого методу, проаналізовано основні переваги та недоліки методу, висунуто пропозиції щодо його покращення.

Ключові слова: відмовостійкість, надлишковий код, латинський квадрат