UCT JOURNAL OF RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY 2019(03)

Available online at http://journals.researchub.org

Implementing a new Torus network with intra-chip encryption

Dr. Azath Mubarakali[1*], Dr. Dinesh Mavaluru[2], Dr. Jayabrabu Ramakrishnan[3]

[1]*College of Computer Science, Department of CNE, King Khalid University, Saudi Arabia*
[2]*College of Computing and Informatics, Saudi Electronic University, Saudi Arabia*
[3]*College of Computer Science and Information Technology, Jazan University, Saudi Arabia*

ARTICLE INFO

Article history:
Received 16 March. 2019
Accepted 30 May 2019
Published 12 July. 2019

Keywords:
Chip Network Perfect Recursive Diagonal Torus (PRDT), Torres Network and RC6 Cryptography

ABSTRACT

In recent years, the ad hoc network has emerged as a solution to the challenges of designing high-performance complex systems at the nanoscale. Maintaining information security in complex systems is an important issue. Therefore, this paper presents a new topology based on the Torres network with encryption capability to build internal network interfaces on the chip with tens or even hundreds of processor units considering the need for information security. The main platform for inter-chip communication is the PRDT network (2.1) and the basic encryption algorithm, RC6. Designing a new node and modifying the source and switched source network PRDT (2,1) encrypts the algorithm based on the RC6 accelerated algorithm. Evaluations performed by the synthesis of this method on the FPGA shows that this provides a 21% overhead of approximately 6% increase in hardware resources that cannot be compromised.

1. INTRODUCTION

Communication networks are used for a variety of purposes.

Their boundaries range from connecting different components of a single device (for example connecting the internal units of a chip) to connecting the nodes of parallel large computers [1].

Considering the wide range of applications of interconnection networks, it is very important to pay attention to information security between nodes. The purpose of this paper is to present a new communication platform based on the Torres network for communication between several processors and information decoding in this context. The second and third sections of Torres PRDT network (2.1) and RC6 encryption and how to accelerate this encryption are discussed.

In section 4, we introduce a new Torus PRDT (2.1) network with accelerated RC6 encryption within the chip and discussed in section 5 of Overhead Computing and Conclusion.

2. METHODOLOGY

2.1. PRDT (2, 1) Network Structure:

PRDT (n, r) is a type of RDT in which each node has all links to higher order Torus. In PRDT (2, 1) the prime number is equal to two and the higher Torus are all of the first order.

As a result, each node contains eight links, four of which belong to the zero-order Torus and the other four belong to the first-order Torus [2].

Of course, PRDT (2.1) with a size of 4×4 has no exception, and each node has five links, four of which are zero and one of them is one. A PRDT (2.1) with a size of 4×4 is shown in Fig. 1 and with a size of 8×8 is shown in Fig. 2.

The source nodes of each independent first-order Torus are: (0,0), (1,0), (2,0), (3,0), (0,1), (1,1), (1,2), (1,3)

Only with zero and one order links, the cost of PRDT (2.1) wiring is significantly reduced.

Also, its routing algorithm is simpler than RDT (2,2,1) / α because it no longer needs floatation.

Therefore, PRDT (2,1) is a very promising method for network interconnections in NOCs with tens and even hundreds of nodes [3].

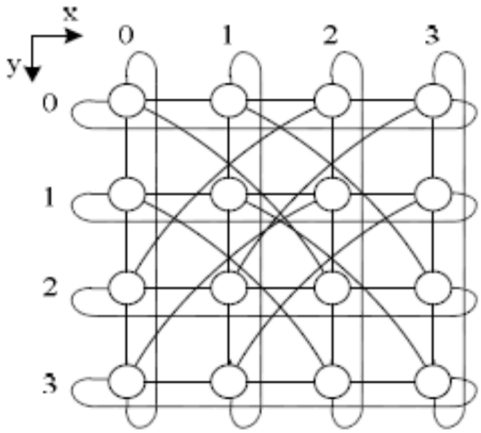

Figure 1. Structure [3] PRDT (2,1) 4 × 4

Figure 2 shows all PRDT links (2.1) with a size of 8×8.

*Corresponding author: mailmeazath@gmail.com
DOI: https://doi.org/10.24200/jrset.vol7iss03pp25-29

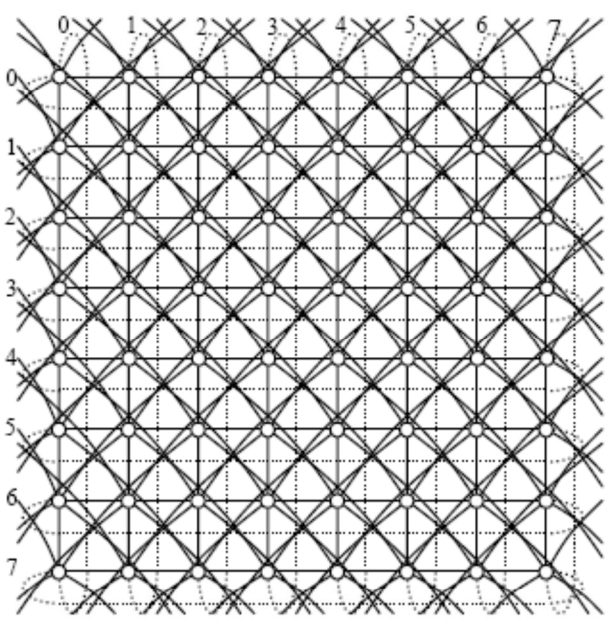

Figure 2. PRDT links (2,1) with size of 8×8

2.2 RC6 encryption

Both the encryption and decryption process are made up of three parts:

The beginning and the end eliminate the possibility of plain text input for the first cryptographic process and for the last cryptographic process.

In the encryption algorithm, registers B and D first undergo changes.

Then a loop is executed on the number of processes in which the registers t and u are first created by rotating the existing function to the size of log (w).

And then registers A and C are obtained by using their XOR with registers t and u and then by rotating the log (w) bit value of t and u and its sum by the key and at the end of the loop, parallel assignment is done.

Finally, the values of the registers A and C are changed [4, 5]. The work of the decryption algorithm is similar to that of the cryptographic algorithm shown in Figure 3 with their pseudo codes.

RC6

Encryption	Decryption
Input: (A, B, C, D) Table S[0..2r+3]	Input: (A, B, C, D) Table S[0..2r+3]

B = B + S[0]
D = D + S[1]
for i= 1 to r do
{
t= (B*(2B+1)) <<< $\log_2 w$
u= (D*(2D+1)) <<< $\log_2 w$
A= ((A⊕t) <<< u) + S[2i]
C= ((C⊕u) <<< t) + S[2i+1]
(A, B, C, D) = (B, C, D, A)
}
A = A + S[2r+2]
C = C + S[2r+3]
	C = C – S[2r+3]
	A = A – S[2r+2]
	for i= r downto 1 do
	{
	(A, B, C, D) = (D, A, B, C)
	u= (D*(2D+1)) <<< $\log_2 w$
	t= (B*(2B+1)) <<< $\log_2 w$
	C= ((C – S[2i+1]) >>> t)⊕u
	A= ((A – S[2i]) >>> u)⊕t
	}
	D = D – S[1]
	B = B – S[0]

Output: (A, B, C, D)	Output (A, B, C, D)

Figure 3. RC6 encryption and decryption algorithm

2.3 Simplification of partial multiplication answer

Given the expression x (2x + 1) in the multiplication formula, it is possible to simplify the partial multiplication result.

To do this, first multiply the expression x by 2x + 1 and obtain the partial product of it.

means: If w is even, then the calculation of f(x) consists of the summation of the partial products of the number of $\frac{w}{2}$, as given in Equation (1).

The above formula allows for the partial product to be automatically generated for each w pair.

According to the algorithm, it is found that after simplification, to calculate the multiplication when x is a 32-bit number, we only need 15 32-bit collectors. [6]

3. RESULTS

3.1Removing carry

It is very difficult to calculate Wallace multipliers normally without the use of a partitioning method for a 32-bit multiplier.

In the above section, we have also concluded that the partial multiplication simplification method cannot be applied to the existing structure by the method of division and solution.

So that better results in terms of delay.

Considering the two above, we conclude that Wallace's multiplication should be implemented in a simpler way, so that Wallace's 32-bit numbers can be used to simplify the partial coefficients.

First, we obtain the result of the partial multiplications as described in the partial multiplication calculation based on the simplification of partial multiplications.

After obtaining the result of simplified partial multipliers for the expression x (2x + 1), we will have 15 32-bit numbers.

These numbers must be added together.

If we consider Fig. 4 as a block, we use Fig. 5 to sum up the numbers, or in other words, to obtain the final result.

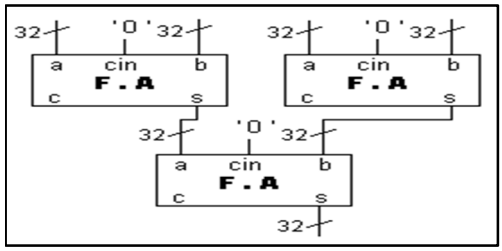

Figure 4. A collector blocker with 4 inputs and a 32-bit output

As shown in the figure, all cores of all 32-bit collectors are omitted. The reason for this is that we only need 32 bits of output at low output.

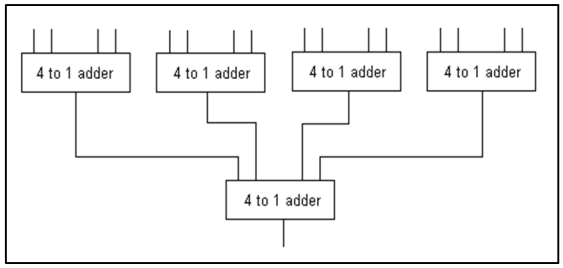

Figure 5. How to make a carry remover multiplier

3.2 Implementation of RC6 algorithm on Torres PRDT network (2.1):

To design the network, first design the network node, then connect the nodes to the entire network.

The network node has a source and a switch, with the network switch having queue, multiplexer and switch controller, each of which must be designed separately.

The components are designed as shown in Figure 6:

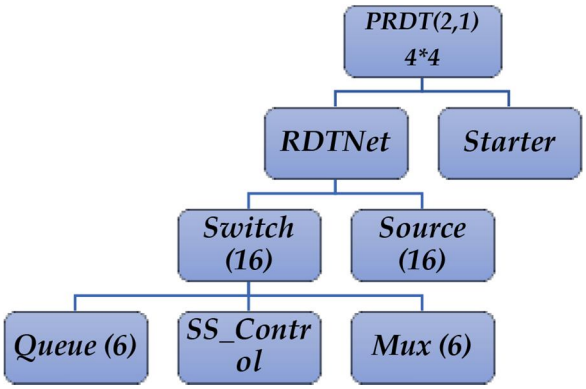

Figure 6. The structure of the Torres network in simulation

4. DISCUSSION

4.1 Source or network source

Cryptography and decryption are performed in this section.

If the source is called as the source, the cryptographic operation is performed.

The following figure illustrates the relationship of RC6 and the source to each other in the more expressive form.

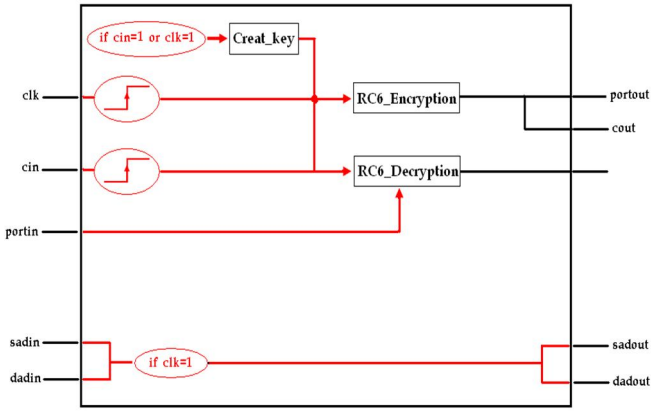

Figure 7. How to connect RC6 and source

4.2 Network switch structure

Each 4x4 PRDT (2.1) grid has six queues, six quasi-multiplexers, and one controller, which you can see in Fig. 8 in the switch you can see how they fit.

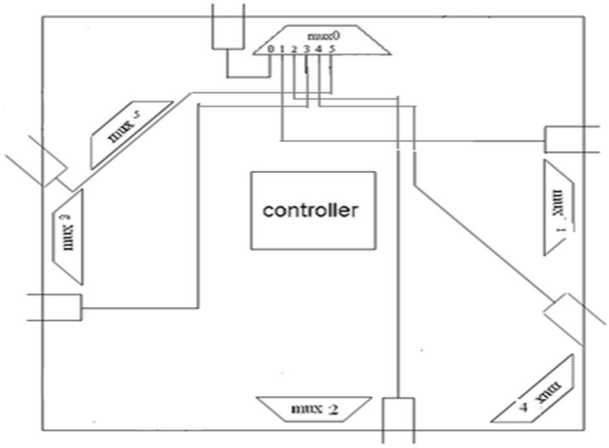

Figure 8. Switch structure

4.3 Network switch queue or buffer

In a switch there are six queues, five of which receive their inputs from the outputs of adjacent switches and a queue is connected to the source and receives its inputs from the source.

So, we define two types of queue for a switch.

One type for the queue connected to the switch outputs and the other type for the queue connected to the source outputs.

Figure 9 illustrates how links are numbered and Figure 10 simulates the whole system.

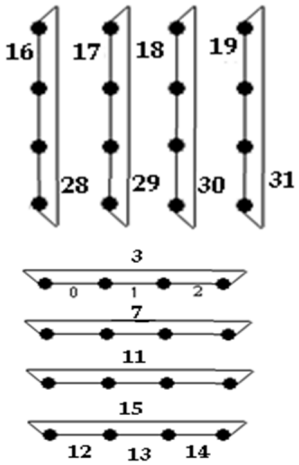

Fig. 9 - Links of the order zero and x and y

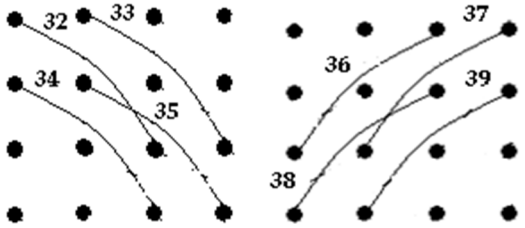

Figure 10. First-rank links with x and y

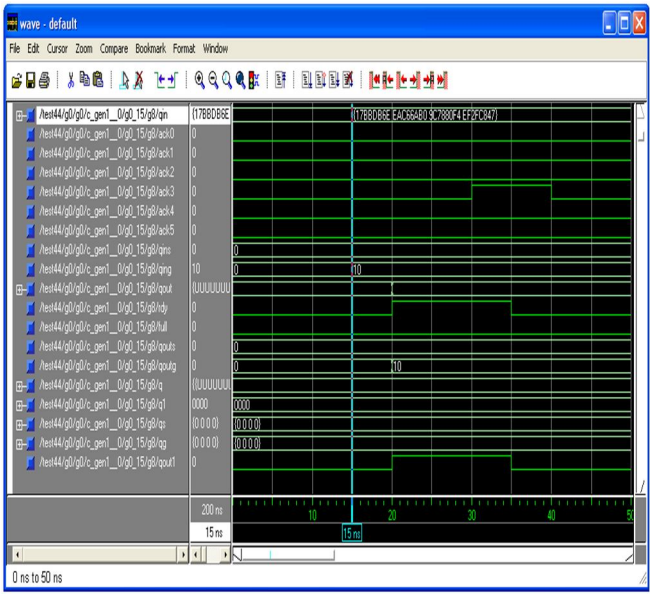

Figure 11. Encrypted data in the switch source queue

4.4 Calculation of overhead caused by RC6 encryption algorithm on Torus network:

The purpose of this section is to investigate the effect of RC6 encryption algorithm on Torres network, to overcome the overhead of this algorithm on the Torres network against the information security. This overhead consists of two parts as follows.

Data overload:

The simple data size of the input and the encrypted data in the RC6 algorithm are the same.

So, the data overhead will only be due to the input key (max. 32 bits) which is very small against the amount of data transferred.

Time Overhead:

To investigate the time overhead caused by RC6 encryption algorithm on Torres network, we investigate the existence and absence of encryption algorithm.

The best and most accurate way is to get a real Synthesis is delay.

Delays and the number of gates consumed in the presence of a network encryption algorithm and its absence in the following tables.

Table 1. Component Consumption Resources without the RC6 Encryption Algorithm

Component Name	source	node	Rdtnet
Number of Slices	27	2597	1438
Number of Slice Flip Flop	46	1912	1297
Number of 4 input LUTs	-	4358	2402
Number Of bonded IOBs	123	613	3140
Number of GCLKs	2	16	16

Table 2. Component Consumption Resources with RC6 Encryption Algorithm

Component Name	source	node	rdtnet
Number of Slices	148	2742	1583
Number of Slice Flip Flop	46	1912	1297
Number Of 4 input LUTs	256	4613	2655
Number Of bonded IOBs	123	613	3140
Number of GCLKs	2	16	16

As shown in the tables, the total resources consumed in the implementation increased by only 6% compared to non-cryptography.

Table 3. Sum of the Delays Obtained for the Components in the RC6 Cryptography

Component Name	With RC6 encryption algorithm	Without RC6 encryption algorithm	Difference between two states
Node	204.806 ns	186.352 ns	18.454 ns
Rdtnet	110.677 ns	77.648 ns	37.66 ns

As shown in Table 3, the latency of the encryption causes a 21% increase in latency.

5. CONCLUSION

The simulation and synthesis results show that, overall, very few headaches, 6% hardware resources increase, and 21% delay in maintaining information security that, it is very acceptable because of this delay is calculated with no encryption and information security.

REFERENCES

[1] Khan, S., Anjum, S., Gulzari, U. A., Umer, T., & Kim, B. S. (2017). Bandwidth-constrained multi-objective segmented brute-force algorithm for efficient mapping of embedded applications on NoC architecture. IEEE Access, 6, 11242-11254.

[2] Yang, Y., Funahashi, A., Jouraku, A., Nishi, H., Amano, H., & Sueyoshi, T. (2001). Recursive diagonal torus: an interconnection network for massively parallel computers. IEEE Transactions on Parallel and Distributed Systems, 12(7), 701-715.

[3] Punhani, A., Faujdar, N., & Kumar, S. (2019). Design and Evaluation of Cubic Torus Network-on-Chip Architecture. International Journal of Innovative Technology and Exploring Engineering, 8(6), 1672-1676.

[4] Khan, S., Anjum, S., Gulzari, U. A., Afzal, M. K., Umer, T., & Ishmanov, F. (2018). An efficient algorithm for mapping real time embedded applications on NoC architecture. IEEE Access, 6, 16324-16335.

[5] Ghafoor, S., Boujnah, N., Rehmani, M. H., & Davy, A. (2019). MAC protocols for terahertz communication: A comprehensive survey. arXiv preprint arXiv:1904.11441.

[6] Khan, S., Anjum, S., Gulzari, U. A., & Torres, F. S. (2017). Comparative analysis of network-on-chip simulation tools. IET Computers & Digital Techniques, 12(1), 30-38.

[7] Saxena, S., Manur, D. S., Mansoor, N., & Ganguly, A. (2020). Scalable and energy efficient wireless inter chip interconnection fabrics using THz-band antennas. Journal of Parallel and Distributed Computing, 139, 148-160.

[8] Gao, Z., Wu, L., Gao, F., Luo, Y., & Zhang, B. (2018). Spoof plasmonics: from metamaterial concept to topological description. Advanced Materials, 30(31), 1706683.