ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.11, No.13, 1–8

Modified TACIT algorithm based on 4H-key distribution for secure routing in NoC architecture

Arun Ambashanker[1a)] **and Palanisamy Nirmal Kumar**[2]

[1] *Anna University, Chennai, India*

[2] *Department of ECE, Anna University, Chennai, India*

a) *aarun.research@gmail.com*

Abstract: The shortcoming of conventional bus architecture has been overcome by adopting Network on Chip (NoC) in multi core System on Chip (SoC). Many works have been anticipated for NoC, based on network topology, switching methodology and routing techniques. This paper proposes a Modified TACIT algorithm for secure routing in NoC architectures. The key generation is based on four types of HASH function (4H-key) that find hard hitting to the intruder. The algorithm is realized under various block sizes as the 8-bit, 64-bit and 128-bit. The device utilization of the proposed algorithm is determined and the results show that, the maximum frequency achieved by the proposed algorithm (for 8-bit block size) is 640.968 MHz with the total memory usage of 78045 Kb.

Keywords: System on Chip (SOC), Network on Chip (NOC), hash function, routing

Classification: Integrated circuits

References

[1] L. Benini and D. Bertozzi: IEE Computers and Digital Techniques **152** (2005) 261. DOI:10.1049/ip-cdt:20045100

[2] P. Pratim Pende, C. Grecu, M. Jones, A. Ivanov and R. Saleh: IEEE Trans. Comput. **54** (2005) 1025. DOI:10.1109/TC.2005.134

[3] P. Guerrier and A. Greiner: DATE '00 (2000) 250. DOI:10.1109/DATE.2000.840047

[4] P. P. Pande, C. Grecu, M. Jones, A. Ivanov and R. Saleh: IEEE Trans. Comput. **54** (2005) 1025. DOI:10.1109/TC.2005.134

[5] B. S. Ferro and P. P. Pande: IEEE Trans. Comput. **58** (2009) 32. DOI:10.1109/TC.2008.142

[6] Y. U. Ogras and R. Marculescu: Proc. Design, Automation and Test in Europe Conference (2007) 1096. DOI:10.1109/DATE.2007.364440

[7] A. Shahabi, N. Honarmand, H. Sohofi and Z. Navabi: IEICE Electron. Express **4** (2007) 332. DOI:10.1587/elex.4.332

[8] S. Saeidi, A. Khademzadeh and A. Mehran: ISSCS 2007 (2007) PID360581. DOI:10.1109/ISSCS.2007.4292678

[9] M. Palesi, R. Holsmark, S. Kumar and V. Catania: IEEE Trans. Parallel Distrib. Syst. **20** (2009) 316. DOI:10.1109/TPDS.2008.106

DOI: 10.1587/elex.11.20140352
Received April 15, 2014
Accepted May 1, 2014
Publicized June 12, 2014
Copyedited July 10, 2014

[10] H. Moussa, A. Baghdadi and M. Jezequel: Proc. IEEE International Symposium on Circuits and Systems (ISCAS) (2008) 97. DOI:10.1109/ISCAS.2008.4541363
[11] Y. H. Kao and H. J. Chao: Proc. 5th Conf. IEEE/ACM International Symposium on Networks on Chip (2011) 81.
[12] P. Gope, A. Singh and A. Sharma: ICECT **5** (2011) 359. DOI:10.1109/ICECTECH.2011.5942020
[13] Y. S. Jeong and S. E. Lee: IEICE Electron. Express **10** (2013) 20130699. DOI:10.1587/elex.10.20130699

1 Introduction

The prolong evolution in Very Large Scale Integrated (VLSI) System has paved the way to System on Chip (SoC) [1]. The immense level of integration makes SoC persistent in domains varies from smart phone, consumer electronics, biological applications, weather forecasting. The SoC consists of various components [2] such as processor, memory unit, I/O interfaces, peripheral, system bus and processing elements. In general, the system bus is used to connect various cores and processing element in Multi-core SoC. The system bus is the bottleneck due to the augment in number of core and processing element. To overcome the issue of the wire-delay problem, Network on Chip (NoC) [3] is introduced. It has several advantages over bus architecture, namely, low power consumption, low latency and higher scalability. The NoC architecture follows set of protocols to communicate between cores. It should posses a specific topology, secure and power aware routing scheme and low power and reliable switching models.

2 Related works

The NoC offers several gains over conventional bus architecture researchers were contributed to quite a lot of NoC architecture [3, 4]. The NoC architecture based on topology, in particular 3D-Mesh topology and stacked mesh is proposed in [5]. In similar, switching in NoC architecture also consider for low power and high speed. A well-known, wormhole switching is proposed in [6]. Routing is considered as the final protocol in NoC architecture. The most familiar routing technique is power-aware routing. In [7] routing based on programmable routing tables is proposed for faulty link in NoC. Three Step Routing algorithms are introduced in [8] for reducing the power consumption during communication. The open problem of Task routing is addressed in [8]. Dead-lock free routing and adaptive routing to reduce the latency is projected in [9]. The complex Dynamic Routing protocol [10] has been used to shun the conflict data buffering. The Output contention problem is resolved using the deflection routing as proposed in [11].

Many works have been involved towards the optimized and efficient routing algorithm for NoC architecture. The contribution varies from low power, buffer contention, high speed, low latency and dead-lock free NoC architectures. But the security of the routing packet is not addressed in the

DOI: 10.1587/elex.11.20140352
Received April 15, 2014
Accepted May 1, 2014
Publicized June 12, 2014
Copyedited July 10, 2014

existing schemes. If the data transferred is not protected means, it will be vulnerable to attacks. A TACIT algorithm has been proposed in [12] for secure data transmission in NoC architecture (As mentioned in our problem statement above). However, the key selection based on HASH function and key distribution is not hard-hitting to the attacker, as well as the bit-reverse process (STEP-6) in the TACIT algorithm [12] can be easily predictable to the attackers. This paper presents a Modified TACIT algorithm based on 4H-key distribution (4-types of HASH function) for secure routing of data in NoC. The invader finds difficulty in breaking the key of a Modified TACIT algorithm due to four types of HASH function used to generate the key.

3 Proposed key generation scheme

The key generation is a difficult task in making the data secure from the invaders. The two types of key available are symmetric and asymmetric key. In asymmetric key cipher, the transmitter and the receiver have different keys for encryption and decryption of the data. On the other hand, in symmetric key both the user has same keys. In [12] the key generation is based on the hash function shown in the hash table. After finding the values of 'p' and 'q', the key is determined by calculating the least prime number between p and q. In this paper, the hash table is modified for four various conditions (4H-key) and tabulated in Table I. In hash function, m, n, u, v denotes the number of lower case alphabetic character, numerical character, upper case alphabetic character and special character. To complicate the key-breaking mechanism, four conditions were checked for generating the key, namely; (i) $mg = m > (n,u,v)$ (ii) $ng = n > (m,u,v)$ (iii) $ug = u > (m,n,v)$ (iv) $vg = v > (m,n,u)$. The algorithm to choose the hash function is shown in Fig. 1. Here, mg represent that the random sequence has more number of lower case alphabetic characters than the other. Initially, a random sequence is generated at the both the routers let it be 'x' and 'y'. Then x and y are exchanged between the router. Here, the first value in the random sequence generated, denotes the value of 'n'.

Table I. Hash table for 4H-key

n	Hash Functions			
	H1	H2	H3	H4
	$m_g = m > (n,u,v)$	$n_g = n > (m,u,v)$	$u_g = u > (m,n,v)$	$v_g = v > (m,n,u)$
0	m^n-m.n	n^u+n.u	u^v+u.v	v^m+v.m
1	m^u+(m+u)	n^v+(n+v)	u^m+(m+v)	v^n+(u+v)
2	m^v-(u+v)	n^m-(m+n)	u^n-(u+n)	v^u-(v+m)
3	n^u+(v.m)	m^v+(u.n)	u^n+(v.m)	v^m+(v.m)
4	n^v+(n.m)	m^u+(u.v)	u^n+(v.n)	v^m+(n.u)
5	n^m-m	m^n-n	u^v-v	v^u-v
6	u^m-m	v^n-n	m^u-u	n^v-v
7	u^n+(n+m-u)	v^m+(m+n-v)	n^v+(v-n-m)	m^u+(u-m-n)
8	u^v+(n+m+v-u)	v^u+(v+u+m-n)	m^n+(m+n+v-u)	n^m+(m+u+n-v)
9	m.n.v+(m.u)	n.u.v+(n.v)	m.n.u+(u.v)	u.v.n+(v.m)

DOI: 10.1587/elex.11.20140352
Received April 15, 2014
Accepted May 1, 2014
Publicized June 12, 2014
Copyedited July 10, 2014

This 'n' is used to find the 'p' and 'q' using the hash functions in the table. In [12], after finding 'p' and 'q' the key value is predicted by least prime number which is easily predictable by trial and error method. In this approach the key value is the average of least and larger prime number between the 'p' and 'q'. Then the generated key is used to cipher the data which is shown in the proposed modified TACIT algorithm.

Hash Function Algorithm	
1	***:Generate*** *(x&y)*
2	***:Exchange*** *x&y*
3	***:If*** *$x=y=m_g$* ***then***
4	***:get*** *'n'; P= H_1 ; Q= H_1;*
5	***:else if*** *($x=y=n_g$)* ***then***
6	***: get*** *'n'; P=H_2;Q=H_2;*
7	***: else if*** *($x=y=u_g$)* ***then***
8	***: get*** *'n'; P=H_3;Q=H_3;*
9	***:else if*** *($x=y=v_g$)* ***then***
10	***: get*** *'n'; P=H_4;Q=H_4;*
11	***:else*** *default*
12	***:end if***
13	***:end***
m_g : Greater no.of lower case Alphabatic Character	
n_g: Greater no.of Numeric Character	
u_g: Greater no.of Upper case Alphabatic Character	
v_g: Greater no.of Special Character	

Fig. 1. Hash function algorithm

4 Proposed modified TACIT algorithm

The encryption algorithm of the proposed Modified TACIT algorithm is shown in Fig. 2. Initially, the permutation process is performed by shuffling the packets.

The ASCII value of the first character is determined using the standard ASCII table. The key generated by 4H-key function is used to perform the XOR operations. Now the TACIT logic (n^k XOR k^k) is executed.

Convert the obtained value from the preceding step into binary form. Here the modification is done in the TACIT logic by carrying out a bit XOR operation and then bit reverse operation. Whereas in [12] the bit reverse operation is executed, this is easily predictable to the attackers. Now the decimal value of the previous step is found out, which will result in the cipher text of the character involved. This process is continued till all the character in the packet is enciphered. In contrast, the decryption process is initiated by finding the decimal value of the first character in ciphered packets. Then the inverse of our modified logic should be executed and then the inverse TACIT logic is applied as shown in Fig. 3. Then the character is determined by reshuffling and this process is continued till all the packets are deciphered.

DOI: 10.1587/elex.11.20140352
Received April 15, 2014
Accepted May 1, 2014
Publicized June 12, 2014
Copyedited July 10, 2014

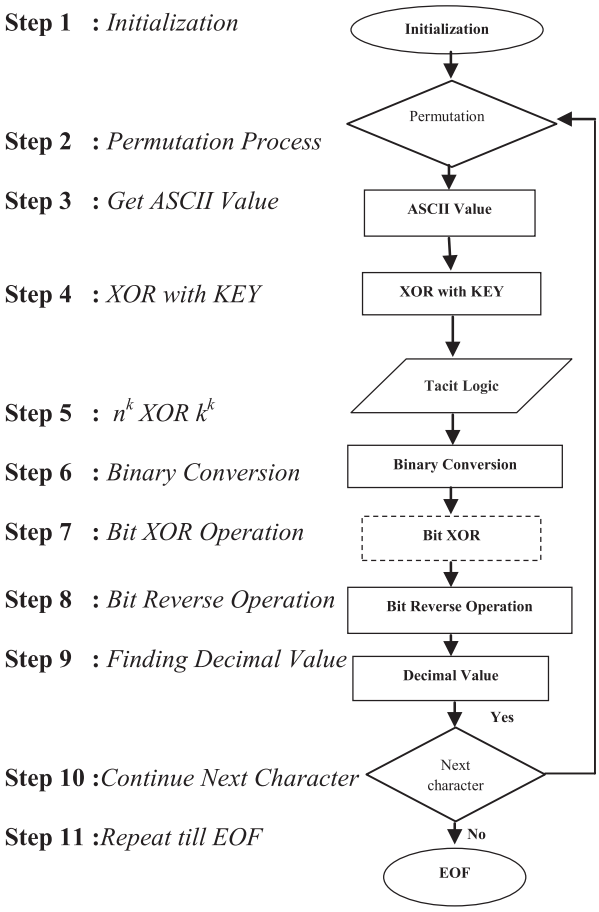

Fig. 2. Proposed encryption algorithm

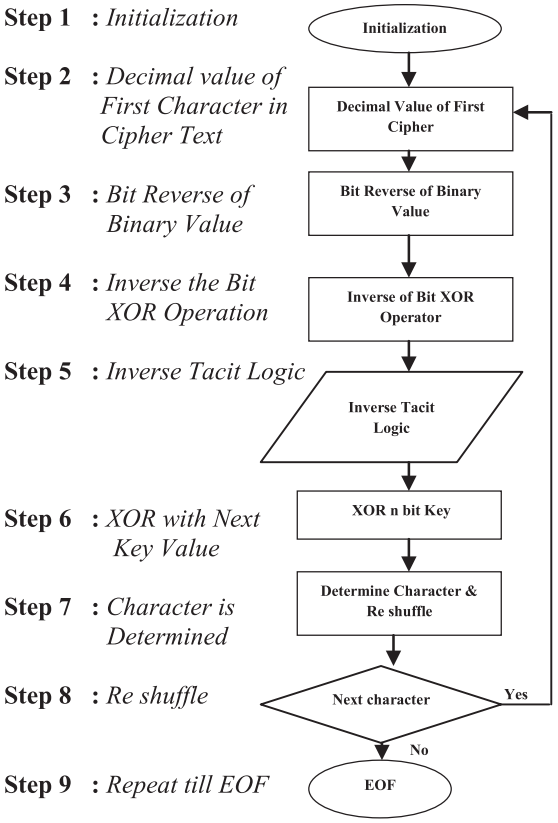

Fig. 3. Proposed decryption algorithm

DOI: 10.1587/elex.11.20140352
Received April 15, 2014
Accepted May 1, 2014
Publicized June 12, 2014
Copyedited July 10, 2014

5 Implementing modified TACIT in NoC architecture

The Network on Chip (NoC) is evolved as a successor to the conventional bus architectures. In NoC, a set of protocol is utilized to enhance the performances. It should follow certain topology, specified routing and switching methods. In this work, X-Y routing algorithm is utilized, since this routing technique [13] is free from the occurrence of deadlock. The Router architecture of NoC is shown in Fig. 4.

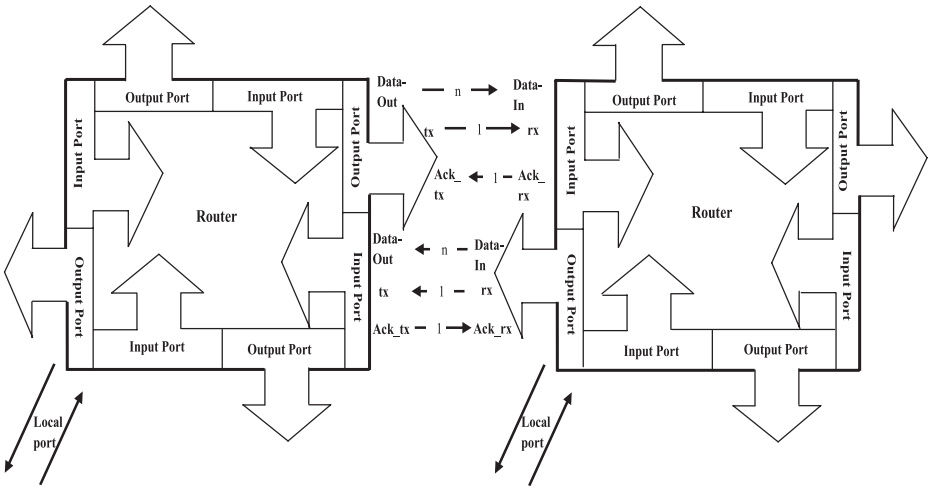

Fig. 4. NoC router architecture

To maintain the flow control this paper considers the acknowledgement signal from the router on either side. Whenever there is a need for transmission of data through output port, the tx is set to '1' and wait till the Ack_tx to be '1', in similar the input-port has to be hang around till Ack_rx to be '1'. To accomplish the security of the data transmitted using the proposed algorithm, the encryption using the modified TACIT is performed in the router and then port availability is checked. The enc function is satisfied means; the router makes tx as '1' and wait for the Ack_tx to be '1', if it becomes '1' then transmit packet through that port. The flow of packet as shown in NoC architecture is illustrated in Fig. 5.

Secure Routing Algorithm

```
1  : function enc
2  : for i ← 0 to 1 do
3  : check for enc
4  : if (enc)
5  : then tx=1
6  : wait for (Ack_tx=1)
7  : if(Ack_tx)
8  : then (Transmit Pkt to Port i)
9  : else go to for loop
10 :  else (Execute enc)
11 :  end enc
12 : end for
13 : end if
14 : end if
```

Fig. 5. Proposed secure routing algorithm

DOI: 10.1587/elex.11.20140352
Received April 15, 2014
Accepted May 1, 2014
Publicized June 12, 2014
Copyedited July 10, 2014

6 Result and discussion

The FPGA realization of the proposed modified TACIT algorithm for NoC architecture is evaluated and the results have been analyzed. The 4H-key generation scheme is used in this work which is illustrated in Appendix A. This representation is recognized with Verilog HDL and Modelsim simulation platform is used to extract the simulation results. The Xilinx FPGA (2vp2fg256-6) is used to target a speed grade of −6. The device utilization report of the proposed encryption and decryption algorithm is shown in Fig. 6 and Fig. 7. Respectively. The tabulated values show the synthesis report for encryption and decryption algorithm for variable block size of 8-bit, 64 bit, 128-bit.

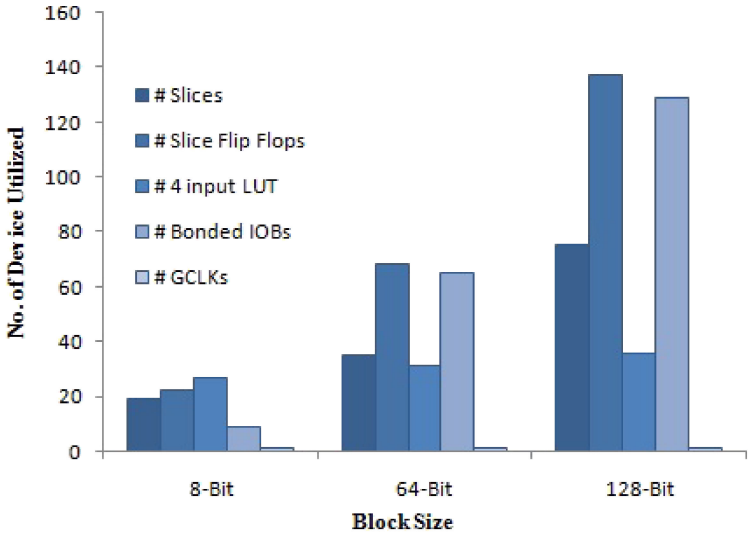

Fig. 6. Device utilization-encryption

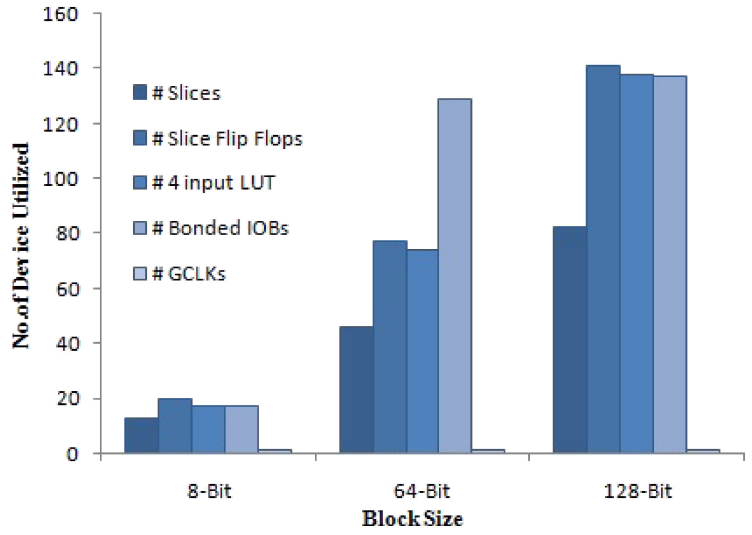

Fig. 7. Device utilization-decryption

The maximum frequency and total memory usage of the proposed algorithm for variable block size is shown in Table II. The proposed algorithm achieves a maximum frequency (for 8-bit block size) of 640.698 MHz and 631.591 MHz for encryption and decryption respectively. In similar, the

DOI: 10.1587/elex.11.20140352
Received April 15, 2014
Accepted May 1, 2014
Publicized June 12, 2014
Copyedited July 10, 2014

Table II. F_{max} and total memory usage

Device Elements	Device Utilization					
	Encryption			Decryption		
	Block Size			Block Size		
	8-Bit	64-Bit	128-Bit	8-Bit	64-Bit	128-Bit
F_{max} (MHz)	640.968	665.460	661.439	631.591	675.739	675.739
Total Memory Usage (Kb)	78045	82413	88013	76005	77316	79301

proposed algorithm (for 8-bit block size) consumes 78045 (Kb) and 76005 (Kb) of memory for encryption and decryption respectively.

7 Conclusion

This paper presents a modified TACIT algorithm for secure routing in Network on Chip (NoC) architecture. The key generation in the proposed algorithm is based on four different types of HASH function (4H-key) scheme. By adopting this technique, the intruder finds difficulty in breaking the key. The proposed algorithm is developed using Verilog HDL and target on 2vp2fg256-6 FPGA. The device utilization of the proposed algorithm is very much lower when compared to high-end boards (more logic devices). Finally, the throughput and the memory usage were determined and tabulated. In future, the algorithm can be implemented in different NoC architectures.

Appendix A: The process of key generation

Let X = (4mph#*%DH@897^jk$) and
Y = (7ln85JZ41@!60&gt)
For X {m = 4, n = 3, u = 2, v = 6}
X satisfies V_g. Therefore P should use Hash table H4.
Here n= 4.
Therefore P = V^m+ (n.u)
=> $(6)^4$+ (3*2)
=> 1296+6 = 1302.
For Y {m = 4, n = 6, u = 2, v = 3}
Y satisfies n_g. Therefore Q should use Hash table H2.
Here n= 7.
Therefore Q = V^m+ (m+n-v)
=> $(3)^4$+ (4+6-3)
=> 81+7 = 88.
Key:
Least Prime number between 88 & 1302 = 89.
Largest Prime number between 88 & 1302 = 1301.
Key = (Least Prime + Largest Prime)/2
=> (1301+89)/2.
Key = 695.

DOI: 10.1587/elex.11.20140352
Received April 15, 2014
Accepted May 1, 2014
Publicized June 12, 2014
Copyedited July 10, 2014