Data Recovery for NOC Using Dictionary Based Encoding

Arun Kumar Mprala
MTech in VLSI and Embedded systems, MS in Electrical Engineering

Pankaj Khurana
MS in Information Technology

DOI: 10.29322/IJSRP.14.10.2024.p15428

Paper Received Date: 14th September 2024
Paper Acceptance Date: 17th October 2024
Paper Publication Date: 22nd October 2024

***ABSTRACT*----**As silicon technology scales, modern processors and embedded systems are rapidly shifting towards complex chip multi-processor (CMP) and system-on-chip (SOC) designs, comprising several processor cores and IP components communicating via a network-on- chip (NOC). As a side-effect of this trend, ensuring their correctness has become increasingly problematic. In particular, the network-on-chip often includes complex features and components to support the required communication bandwidth among the nodes in the system.

In this work we are proposing a deadlock recovery, by using dictionary based encoding techniques. Normally deadlock is a condition where data loss is a major problematic effect in networks. Previous conditions for deadlock recovery uses a multi-path generally transitive closure methods and technique where packets are allowed to pass through an different path if packet is found in deadlock this leads to increased traffic , complexity and also time requirement is more, this technique fails when more than one packet enters into deadlock. To overcome this we are using a data recovery of packet to ensure the correctness of packet, at the receiver end if the data inside packet is corrupted the data recovery methods are adopted to avoid Network utilization, this technique avoids Re-request of packet from transmitter by receiver and transmitter is not required to resend the same packet again. Hence the network traffic is greatly reduced using this technique.

***Index Terms*—**File, Packet Framing, Deadlock, transceiver, Dictionary Encoding.

I. INTRODUCTION

DATA being transmitted over a network is increasing. Due to the support of high speed networks the data transfer rate and data handling requirement is rapidly increasing. Even more to consider the data handling is done by the networks which includes NOC to maintain reliable communication between nodes from transmitter end and receiver end. The requirement of higher bandwidths in the channel due to higher requirement of user node connections made networks more and more complicated as each node end points has to be connected to each other node with high end security and reliability in data also. The network traffic also has increased as data packets flows between nodes. On an average 60%of the networks packets are due to re-request of the packet from the receiver end point to the transmitter.

A deadlock is a situation in NoCs wherein two or more packets are waiting for one another to release channels, and are unable to make progress. The network is dead- locked if a chain of waiting packets forms a cycle. As a simple deadlock example, consider the dependency cycle of four packets shown in Fig. 1. Each of the four packets in the figure holds some channels but it cannot proceed further until it acquires another channel currently held by one of the rest of the packets. However, no packet can release the channel needed by another until it acquires its own requested channel. The packets are deadlocked and will remain in this state halting all the occupied channels until some intervention brings them back to life by breaking the dependency cycle. Deadlocks can paralyze network communications by halting the resources occupied by the deadlocked packets which in return could increase other packets blocking [3]. Hence it is crucial to remove deadlocks. There are two strategies to deal with deadlocks: avoidance and recovery [4], [5]. In deadlock avoidance, resources are granted to packets in a way that the overall network is deadlock free. This can be based on a turn-model which prohibits the routing algorithm from making certain turns in the network [6], [8] or based on the strict ordering of virtual channels [4]. In general, avoidance techniques require restricted routing functions or additional resources to prevent deadlocks. Due to its simplicity the turn-model is popular in NoCs, even though it limits the routing alternatives and diminishes fault tolerance capabilities.

Also the turn-model is not applicable to an arbitrary network topo1ogy [12]. Alternatively, deadlock recovery implies that resources are granted to packets without any routing restrictions, potentially outperforming deadlock avoidance [7], [9], [10], [11]. Deadlocks may occur and efficient detection and recovery mechanisms are required to intervene. However detecting deadlocks in a network is challenging because of their distributed nature

The packet loss occurrence is due to the deadlock conditions at the network chips caused by the heavy packet traffic being delayed to be switched. Other causes are due to the data recovery. Recover data properly and requests for the resend of the packet again this leads to rapid increase in the network traffic.

The data recovery for NOC technique is an efficient promising approach for network communication which recovers the packet when found in deadlock and maintains the correctness of the packet, at the receiver end data inside the packet is verified if the data is corrupted the original data is recovered using decoding techniques.

Related Work

The first packet framing at the transmitter end was proposed by Kenichi Mori, Abderazek Ben Abdullah, and Kenichi Kuroda [13]. Their technique uses packet design for switching and routing at faster speeds and reliable communication being achieved. The network unit is placed between transmitter and receiver module to understand the base of connections in a network. The encoding techniques used here include the general methods adopted for IPv4 checksum; another Encoding method uses a binary compression with encoding using approach of Bit mask code compression with dictionary selection methods.

The techniques discussed so for includes the packet design, Deadlock detection and packet data recovery is being adopted using deadlock detection and correction methods, this recovery includes the same decompression methods adopted at the deadlock detection end and at the receiver module. The proposal of data recovery is considered only on the basis of efficient data recovery which requires best encoding technique. Several techniques have been proposed to improve the standard dictionary-based code compression by considering mismatches. The basic idea is to create repeating patterns from mismatches by storing the differences during code compression. We perform detailed analysis of these techniques to outperform our technique.

DATA RECOVERY

We have developed an efficient data recovery algorithm for NOC devices, it has four major steps:1) Packet design where packet is framed using standard packet design as shown below with minor required changes; 2) Deadlock detection and packet data correction; 3) Receiver module which reconstructs the data and avoids re request using best effort decompression logic. 4) Encoding which includes data encoding to maintain efficient communication.

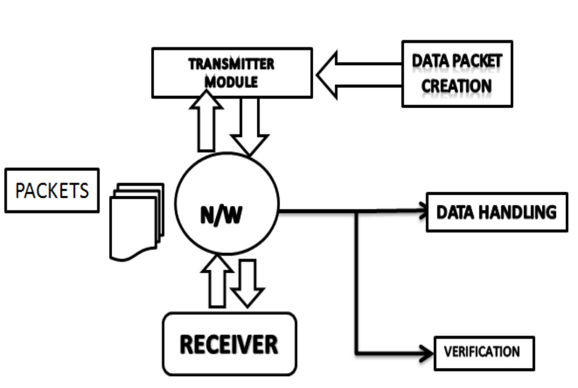
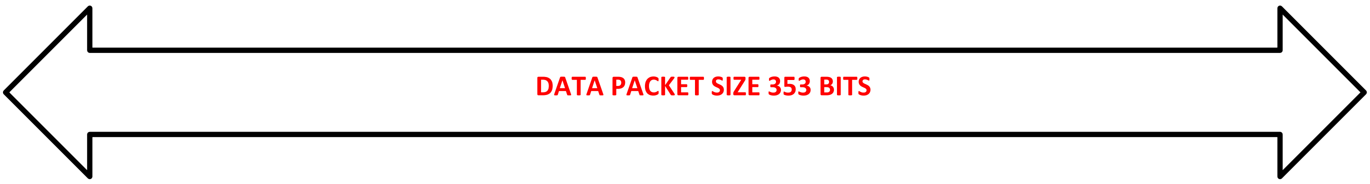


FIG:1 :WORKING BLOCK Diagram

A: PACKET DESIGN

START	ED	SIP	DIP	CRC	PK	SMAC	DMAC	PORT	DATA	STOP
1	32	32	32	32	32	48	48	32	64	1
S	D	S	S	D	D	S	S	S	D	S



FIELD	ABREVATION	DESCRIPTION	TYPE
START	START	Start of the packet	STATIC
ED	Encoded data	Encoded data described from source data	DYNAMIC
SIP	Source IP	Source IP address of packet	STATIC
DIP	Destination IP	Destination IP address of packet	STATIC
CRC	Check sum	Check sum value of packet	DYNAMIC
PK	Packet info	Frame details and packet number	DYNAMIC
SMAC	Source MAC	Source mac address of packet	STATIC
DMAC	Destination MAC	Destination MAC address	STATIC
PORT	PORT NUMBERS	Indicates application port number	STATIC
DATA	DATA FIELD	Actual data is loaded.	DYNAMIC
STOP	STOP BIT	End of packet	STATIC

B. DEADLOCK DETECTION

Deadlock detection is done by considering the input packet average value when the minimum percentage of packet is not received within a mean time; deadlock data recovery options are applied to verify the correctness of the packet data.

When a packet is found in deadlock the packet is analyzed to verify that minimum packet information is available or not in the received data, if minimum information is not received then the data recovery functions fails to recover packet. Minimum of 30% to 40% of the packet information is required to apply the data recovery else the data recovery methods cannot be applied to the packet.

C.CONDITIONS DURING DEADLOCK DETECTION AND CORRECTION

Case1: PS<30%

DEF: Input packet size (PS) has <(less than) 30% information.

Data recovery cannot be applied hence the packet needs to be resent from the transmitter end; packet will be made crashed such that the packet will not disturb the receiver.

Case2: PS>30 %< 70%

DEF: Input packet information > (greater than) 30% <(less than) 70%.

Packet is considered to be in deadlock, packet data recovery methods are applied to reconstruct the packet back again the fields of MAC ADDRESSES, PORT NUMBERS, PACKET INFORMATION is replaced with zeros and the original data in data field is recovered.

D. Responder

The responder is an data receiving unit which receives the input packet verifies the validity of packet data and when found to be corrupted using the data field values of Encoded data, check sum and packet information the data is corrected .The decoder plays an very important role in the receiver end for the data correction. Finally the data from the packet is extracted and copied to a text file for validation.

F. Encoding

Encoding is done using dictionary based along with bit mask to achieve efficient compression, here the input data is considered to be Binary data of variable length, hence compression requires a moderate complex method such as dictionary based for maintaining packet correctness and data correctness inside the packet.

RESULTS

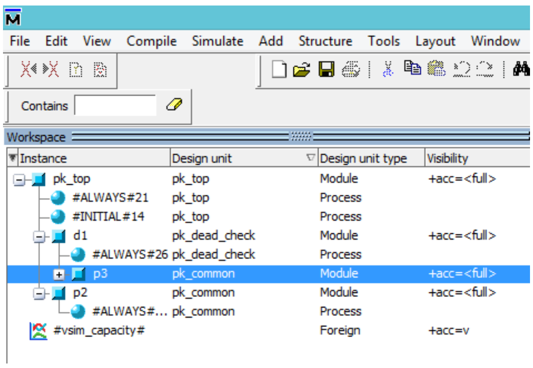


FIG 4.1. HIERARCHY OF RTL DESIGN

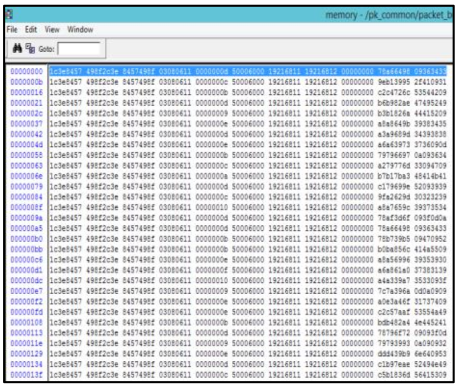
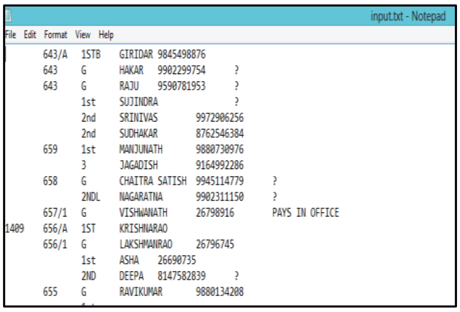
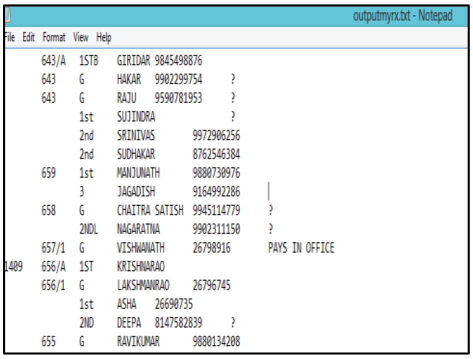


FIG 4.2 Allocated memory location for PACKET BUFFER.



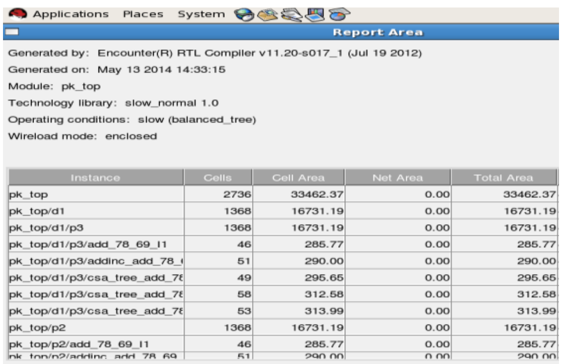

```
        643/A   1STB    GIRIDAR 9845498876
        643     G       HAKAR   9902299754      ?
        643     G       RAJU    9590781953      ?
                1st     SUJINDRA                ?
                2nd     SRINIVAS        9972906256
                2nd     SUDHAKAR        8762546384
        659     1st     MANJUNATH       9880730976
                3       JAGADISH        9164992286
        658     G       CHAITRA SATISH  9945114779      ?
                2NDL    NAGARATNA       9902311150      ?
        657/1   G       VISHWANATH      26798916        PAYS IN OFFICE
1409    656/A   1ST     KRISHNARAO
        656/1   G       LAKSHMANRAO     26796745
                1st     ASHA    26690735
                2ND     DEEPA   8147582839      ?
        655     G       RAVIKUMAR       9880134208
```

FIG 4.3 INPUT DATA FILE




```
        643/A   1STB    GIRIDAR 9845498876
        643     G       HAKAR   9902299754      ?
        643     G       RAJU    9590781953      ?
                1st     SUJINDRA                ?
                2nd     SRINIVAS        9972906256
                2nd     SUDHAKAR        8762546384
        659     1st     MANJUNATH       9880730976
                3       JAGADISH        9164992286
        658     G       CHAITRA SATISH  9945114779      ?
                2NDL    NAGARATNA       9902311150      ?
        657/1   G       VISHWANATH      26798916        PAYS IN OFFICE
1409    656/A   1ST     KRISHNARAO
        656/1   G       LAKSHMANRAO     26796745
                1st     ASHA    26690735
                2ND     DEEPA   8147582839      ?
        655     G       RAVIKUMAR       9880134208
```

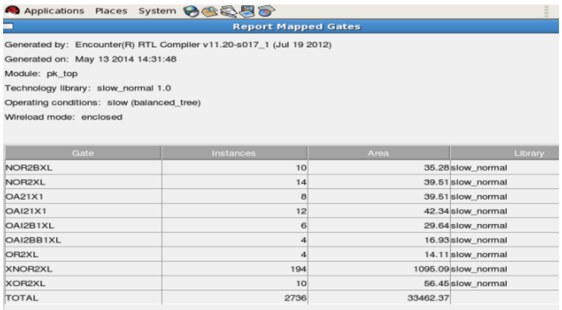
FIG 4.4 Output data file.



Generated by: Encounter(R) RTL Compiler v11.20-s017_1 (Jul 19 2012)
Generated on: May 13 2014 14:33:15
Module: pk_top
Technology library: slow_normal 1.0
Operating conditions: slow (balanced_tree)
Wireload mode: enclosed

Instance	Cells	Cell Area	Net Area	Total Area
pk_top	2736	33462.37	0.00	33462.37
pk_top/d1	1368	16731.19	0.00	16731.19
pk_top/d1/p3	1368	16731.19	0.00	16731.19
pk_top/d1/p3/add_78_69_I1	46	285.77	0.00	285.77
pk_top/d1/p3/addinc_add_78_(	51	290.00	0.00	290.00
pk_top/d1/p3/csa_tree_add_78	49	295.65	0.00	295.65
pk_top/d1/p3/csa_tree_add_78	58	312.58	0.00	312.58
pk_top/d1/p3/csa_tree_add_78	53	313.99	0.00	313.99
pk_top/p2	1368	16731.19	0.00	16731.19
pk_top/p2/add_78_69_I1	46	285.77	0.00	285.77
[illegible]	51	290.00	0.00	290.00

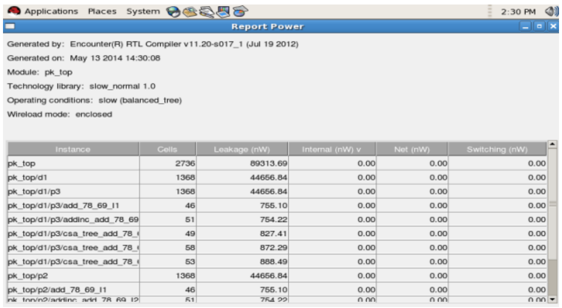
FIG 4.5 Area report.



Generated by: Encounter(R) RTL Compiler v11.20-s017_1 (Jul 19 2012)
Generated on: May 13 2014 14:31:48
Module: pk_top
Technology library: slow_normal 1.0
Operating conditions: slow (balanced_tree)
Wireload mode: enclosed

Gate	Instances	Area	Library
NOR2BXL	10	35.28	slow_normal
NOR2XL	14	39.51	slow_normal
OA21X1	8	39.51	slow_normal
OAI21X1	12	42.34	slow_normal
OAI2B1XL	6	29.64	slow_normal
OAI2BB1XL	4	16.93	slow_normal
OR2XL	4	14.11	slow_normal
XNOR2XL	194	1095.09	slow_normal
XOR2XL	10	56.45	slow_normal
TOTAL	2736	33462.37	

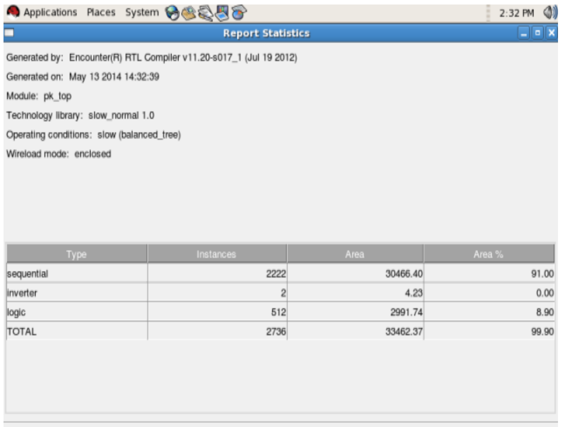
Fig 4.6 Gates report



Generated by: Encounter(R) RTL Compiler v11.20-s017_1 (Jul 19 2012)
Generated on: May 13 2014 14:30:08
Module: pk_top
Technology library: slow_normal 1.0
Operating conditions: slow (balanced_tree)
Wireload mode: enclosed

Instance	Cells	Leakage (nW)	Internal (nW) v	Net (nW)	Switching (nW)
pk_top	2736	89313.69	0.00	0.00	0.00
pk_top/d1	1368	44656.84	0.00	0.00	0.00
pk_top/d1/p3	1368	44656.84	0.00	0.00	0.00
pk_top/d1/p3/add_78_69_I1	46	755.10	0.00	0.00	0.00
pk_top/d1/p3/addinc_add_78_69	51	754.22	0.00	0.00	0.00
pk_top/d1/p3/csa_tree_add_78_(	49	827.41	0.00	0.00	0.00
pk_top/d1/p3/csa_tree_add_78_(	58	872.29	0.00	0.00	0.00
pk_top/d1/p3/csa_tree_add_78_(	53	888.49	0.00	0.00	0.00
pk_top/p2	1368	44656.84	0.00	0.00	0.00
pk_top/p2/add_78_69_I1	46	755.10	0.00	0.00	0.00
[illegible]	51	754.22	0.00	0.00	0.00

Fig 4.7 Power report.



Generated by: Encounter(R) RTL Compiler v11.20-s017_1 (Jul 19 2012)
Generated on: May 13 2014 14:32:39
Module: pk_top
Technology library: slow_normal 1.0
Operating conditions: slow (balanced_tree)
Wireload mode: enclosed

Type	Instances	Area	Area %
sequential	2222	30466.40	91.00
inverter	2	4.23	0.00
logic	512	2991.74	8.90
TOTAL	2736	33462.37	99.90

Fig 4.8 Statistical delay report.

I. *CONCLUSION*

This work studies deadlock recovery strategy in NoCs as opposed to deadlock avoidance. We proposed a new deadlock detection method based on computing data equivalence sets at run-time.. The new method eliminates the need of any kind of timeout mechanism, and delivers true deadlock detections independent of the network load rather than approximating with congestion estimation as in the existing methods.

The network traffic is greatly controlled by data handling and maintaining correctness of the data. This data recovery method reduces network traffic around 40% to 60% on an average.

REFERENCES

[1] L. Benini and G. De Micheli, ”Networks on chips: a new SoC paradigm,”IEEE Computer, vol. 35, no. 1, pp. 70-78, 2002.

[2] W. Dally and B. Towles, ”Route packets, not wires: on-chip interconnection networks,” in Proc. 38th Design Automation Conf., Jun. 2001, pp. 684-689.

[3] S.Warnakulasuriya and T. Pinkston,”Characterization of dead-locks in interconnection networks,” in IPPS 97. IEEE Computer Society, 1997, pp. 80-86.

[4] J. Duato, S. Yalamanchili, and L. M. Ni, Interconnection Networks: An Engineering Approach.Morgan Kaufmann Publishers 2004.

[5] W. J. Dally and B. Towles, Principles and Practices of Interconnection Networks.Morgan Kaufmann Publishers, 2004.

[6] J. G. Christopher and M. N. Lionel, ”The turn model for adaptive routing,” in Proc. 9th Parallel Processing Int. Symp., vol. 41, no. 5, 1994, pp. 874-902.

[7] K. Anjan and T. Pinkston, ”DISHA: a deadlock recovery scheme for fully adaptive

routing,” in Proc. 9th Parallel Processing Int. Symp., 1995, pp. 537-543.

[8] G.-M. Chiu, ”The odd-even turn model for adaptive routing,” IEEE Transac. Parallel Distrib. Syst., vol. 11, no. 7, pp. 729-738, 2000.

[9] J. H. Kim, L. Ziqiang, and A. A. Chien, ”Compressionless routing: a framework for adaptive and fault-tolerant routing,”IEEE Trans. on Parallel and Distrib. Syst., vol. 8, no. 3, pp. 229-244, 1997.

[10] J. M. Martnez-Rubio, P. Lpez, and J. Duato, ”A cost-effective approach to deadlock handling in wormhole networks,” IEEE Trans. Parallel Distrib. Syst., vol. 12, no. 7, pp. 716-729, 2001.

[11] J. M. Martinez-Rubio, P. Lopez, and J. Duato, ”FC3D: Flow controlbased distributed deadlock detection mechanism for true fully adaptive routing in wormhole networks,” IEEE Trans. On Parallel Distributed Syststem, vol. 14, no. 8, pp. 765-779, 2003.

[12] A. Lankes, T. Wild, A. Herkersdorf, S. Sonntag, and R. Reinig, ”Comparison of deadlock recovery and avoidance mechanisms to approach message dependent deadlocks in on-chip networks,” NOCS10, 2010, pp. 17-24.

[13] Kenichi Mori, Abderazek Ben Abdallah, Kenichi Kuroda” Design and Evaluation of a Complexity Effective Network-on-Chip Architecture on FPGA”,p318-321.