ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 6, Issue 6, June 2019

# Error Detection And Correction Between The Router In NOC Using Modified Decimal Matrix Code

Nurul Mubeen K[1], Latha N[2]

M.E. Scholar , VLSI Design, MIET engineering college, Tiruchirapalli, India, nurulmubeen@gmail.com[1]
Assistant Professor, Department of Electronics and Communication Engineering, MIET engineering college, Tiruchirapalli,India,latha.n@miet.edu[2]

**Abstract:** A router architecture is proposed with error detection and correction technique. The routing is performed by xy routing algorithm. Due to the presence of error in transmission through channel whole packet is discarded and it calls for retransmission for which the power and time are wasted. In order to avoid re-transmission, error detection and correction codes are introduced. In the existing system DECIMAL MATRIX CODE is used as error detection and correction technique, in which the redundancy bits are more. In order to reduce the number of redundancy bits MODIFIED DECIMAL MATRIX CODE, the decimal addition is replaced by hamming code, is proposed. By reducing the number of redundancy bits, power and area gets reduced and increases the reliability of transmission. The router with codec is designed and implemented in standard cell design using Mentor Graphics EDA tool.

**Keywords**: Router, Detection, Packets, Redundancy

## I. INTRODUCTION

System on chip help us to put the technology advancements into a single chip and for communication within it we go for network on chip .Within NOC router is used to forward packets which is received from one link along with destination address to another link to send packet to receiver. packets are traversed through channel where error will occur so error correction codes are used .decimal matrix code is the ECC used here. the proposed router architecture with MDMC is designed and implemented.

## II. RELATED WORK

Adaptive routing algorithms needs additional resources to store the information and routing decision gets complexed[1].In order to maintain the routing table the overhead gets increased in congestion aware routing algorithm[2]. Genetic algorithm[3] cannot guarantee an optimal solution, do not scale well with complexity, have a tendency to converge towards local optima or even arbitrary points rather than the global optimum of the problem. In the circuit switching, packets are sent by making physical connections between routers which takes too long time and both should have same data rate while transmitting packet through channel, at any point of time no packet is available for sending then channel gets wasted[4].the data is routed with several layers of encapsulation of security before it reaches destination so delay increases, basically used for transmitting sensitive information so our IP address will come for surveillance even we didn't use for any harmful data sending in the Tor network[5].Error may occur in random manner in real time ,but in triple adjacent error correction technique it can correct errors which is adjacently located only[6]. FIFO is replaced by elastic buffer in the router .Elastic buffer are never intended to be completely empty or completely full and used to align incoming data with a separate reference signal [7]. Hamming code[8] is used to detect and correct one error only and if the packet size is more then this technique is infeasible. In directional routing algorithm, duplicate packets may circulate forever unless we enforce a network topology without loops, each node should track every packet and only forward each packet once [9]. In multi path routing moving to the alternative path will incur a potentially disruptive period during which the connection is re-established[10].

## III. PROPOSED ROUTER ARCHITECTURE

The router top module consist of FIFO, register, synchronizer, FSM. The information is sent through packet switching. The packet has three fields: header, payload, parity. The header has the destination address. Payload has

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 6, Issue 6, June 2019

data information. Parity field contain the result of security check information.

### A. FIFO

The data_in is given when write_enb goes high. FIFO operates with the help of clock and soft reset. Soft reset is given by synchronizer. The write operation is done when FIFO is not full. After data_in is given the write_enb goes low, read_enb goes high and data is obtained at data_out. Empty should be high so that data_in can be given. Extra bit is added in data to predict header byte. An internal counter is added to payload length and its starts reducing till it reaches zero, after that new packet is loaded.

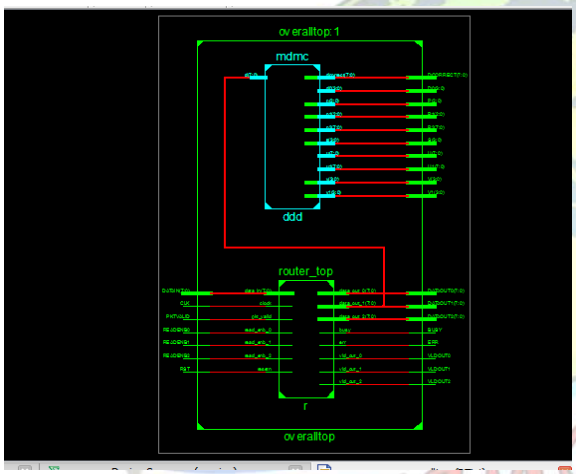


**Fig. 1. Proposed router module with MDMC**

### B. Synchronizer

Synchronizer provides the syncing of FSM and FIFO.A temp register is used to hold the data_in when detect_add signal is high. Based on the data_in, FIFO is selected as follows. If data_in = 00 then data goes to FIFO 0, data_in = 01 then data goes to FIFO 1, data_in = 10 then data goes to FIFO 2. The vld_out X is created when empty X gos low. Write_enb_reg is used to write the data_in in corresponding FIFO. If the vld_out X is high and read_enb X is low below 30 clock cycles then soft_reset X goes high.

### C. Register

- low_packet_valid goes high when ld_stste = 1, pkt_valid = 0.
- If (ld_state = 1, low_packet_ valid = 1 and parity_done = 0) or (ld_state = 1, fifo _full = 0, pkt_ valid = 0) then parity_done = 1.
- If parity_done = 1 it checks for packet_parity-byte = internal_parity_byte then err = 0, otherwise err = 1. packet_parity_byte and internal_parity_byte are internal registers.
- If pkt_valid = 0 and ld_state = 1 then data_in is stored in packet_parity_byte.
- If (lfd_state = 1) or (pkt_valid = 1, ld_state = 1, and full_state = 0) then internal_parity_byte = header_byte ∧ internal_parity_byte.
- If ld_state,fifo full = 1 then fifo_full state is internal register. Data_in is stored in fifo_full state.
- lfd_state = 1, then dout = header byte. ld_state = 1 and fifo full = 0, then dout = data_in. If laf_state = 1 then dout = fifo_full state byte.

### D. There are eight states since we use three FIFO's

**1. Decode_address**

- Detect_add is used to detect the arrival of packet
- If ((pkt_valid & data_in [1;0] = 0 & fifo_empty_0) | (pkt_valid & data_in [1;0] = 1 & fifo_empty_1) | (pkt_valid & data_in [1;0] = 2 & fifo_empty_2)) then it goes to load_first_data.
- If ((pkt_valid & data_in [1;0] = 0 & ~fifo_empty_0) | (pkt_valid & data_in [1;0] = 1 & ~fifo_empty_1) | (pkt_valid & data_in [1;0] = 2 & ~fifo_empty_2)) then it goes to wait_till_empty

**2. Load_first_data**

- lfd_state = 1 lfd_state is used to load data to FIFO Busy = 1.
- Busy is used as the header byte It is not changed to new value for present packet.
- It goes unconditionally to load_data.

**3. Load_data**

- ld_state = 1, which is done to load data to FIFO. Busy = 0, to receive new data for each clock cycle. write_enb_reg = 1, to write data to FIFO.
- If fifo_full = 1, then it goes to fifo_full_state.
- If fifo_full = 1, pkt_valid = 0 then it goes to load_parity.

**4. Load_parity**

Busy = 1, write_enb_reg = 1, it goes unconditionally to check_parity_error.

**5. FIFO_Full_state**

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 6, Issue 6, June 2019

- Busy = 1, full_state = 1. If fifo_full = 0, then it goes to load_after _full.
- If fifo_full = 1, then it goes to fifo_full_state.

**6. Load_after_full**

- Laf_state = 1, busy = 1, write_enb_reg = 1.
- If parity_done = 1, then it goes to Decode_address. If parity_done = 1, then it goes to head_parity.

**7. Wait_till_empty**

- If fifo_empty_X = 0, then it goes to wait_till_empty.
- If fifo_empty_X = 1, then it goes to load_first_data.

**8. Check parity error**

- Busy = 1, rest_int_reg = 1. If fifo_full = 0, then it goes to decode address.
- If fifo_full = 1, then it goes to fifo_full_state.

## IV. PROPOSED MODIFIED DECIMAL MATRIX CODE

### *E. Encoder of MDMC*

- The divide-symbol and arrange-matrix :The *N*-bit word is divided into k symbols of m bits (N=k x m)and these symbols are arranged in a 2-D matrix (k=k1 x k2) where k1= numbers of rows and k2= numbers of columns in the logical matrix.
- the horizontal redundant bits H: Hamming codes for each symbol of first row.
- vertical redundant bits V: binary operation among the bits per column.

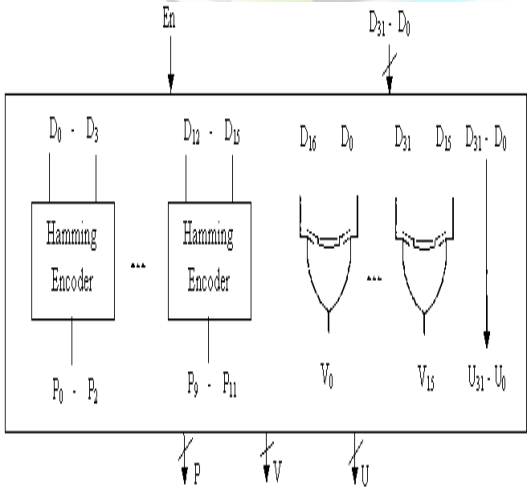


**Fig. 2. Encoder of Modified Decimal Matrix Code**

For example, N= 32-bit ,which is divided into eight symbols of 4-bit. D0 to D31= information bits, P0 to P11 =horizontal check bits, V0 to V15 = vertical check bits.

The following equations represent the Hamming equations to calculate the horizontal check bits.

P0 = D3 ^ D1 ^ D0
P1 = D3 ^ D2 ^ D0
P2 = D3 ^ D2 ^ D1 …….and so on

The following equations are used to compute the vertical check bits.

V0 = D0 ^ D16
V1 = D1 ^ D17 …… and so on

The symbol "^"indicates binary exclusive-OR operation. The remaining bits U31 – U0 are directly copied from D31 to D0. Depending upon the values of k and m, the number of redundant bits varies so it should be carefully changed.

### *F. Decoder of Modified decimal matrix code*

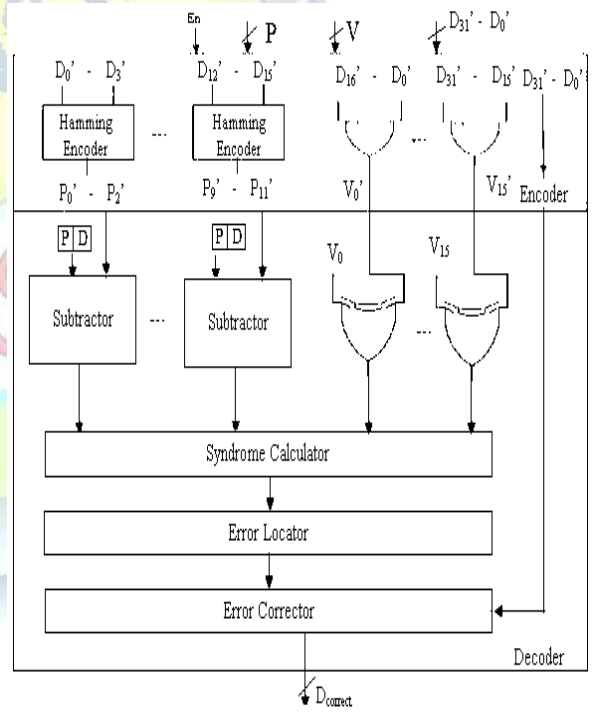


**Fig. 3. Decoder of Modified Decimal Matrix Code using ERT**

In the decoding process, we need to detect the errors if any and correct them accordingly. Here foremost step is using encoder re-use technique. The received information

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 6, Issue 6, June 2019

bits D' are applied to the in-built encoder block in the decoder circuit to obtain the horizontal check bits P0' to P11' and vertical check bits V0' to V15'. The decoding process goes through a step by step process i.e., syndrome calculator, error-locator and error corrector. The decimal integer subtraction is used to compute horizontal syndrome bits and exclusive OR operation to compute vertical syndrome bits. The non-zero horizontal syndrome bits indicates error detection and the non-zero vertical syndrome bits gives the location of errors. These errors are corrected by error corrector block using xor operations.

The horizontal syndrome bits are obtained as follows:

▲ P0P1P2D3D2D1D0 = P0P1P2D3D2D1D0' – P0P1P2D3D2D1D0

▲ P3P4P5D7D6D5D4 = P3P4P5D7D6D5D4' – P3P4P5D7D6D5D4 ….and so on

The vertical syndrome bits are obtained as follows:

S0 = V0' ^ V0 S1 = V1' ^ V1 …….and so on

The errors can be corrected by D0correct = D0' ^ S0 … and so on

**ERROR DETECTION**

1- If ▲P0P1P2D3D2D1D0 ≠ 0 then error occurred in symbol 0 and no error in symbol 1.

2- If ▲P0P1P2D3D2D1D0 = 0 and S3S2S1S0 ≠ 0 then error occurred in symbol 1 and no error in symbol 0.

3- If ▲P0P1P2D3D2D1D0 = 0 and S3S2S1S0 = 0 then no errors in symbol 0 and symbol 1.

Similarly the rest of the errors can be detected.

Here,

▲P0P1P2D3D2D1D0 = P0P1P2D3D2D1D0' - P0P1P2D3D2D1D0

= 0110011 – 1001100

= 1100111 ≠ 0 => error occurred in symbol 0.

**ERROR LOCATION**

S0 = V0' ^ V0 = 0 ^ 1 = 1

S1 = V1' ^ V1 = 0 ^ 1 = 1

S2 = V2' ^ V2 = 1 ^ 0 = 1

S3 = V3' ^ V3 = 1 ^ 0 = 1 …..and so on

**ERROR CORRECTION**

D0 correct = D0' ^ S0 = 1 ^ 1 = 0

D1 correct = D1' ^ S1 = 1 ^ 1 = 0

D2 correct = D2' ^ S2 = 0 ^ 1 = 1

D3 correct = D3' ^ S3 = 0 ^ 1 = 1……and so on

**Syndrome Calculator**:the received redundant bits and the original redundant bits is compared to get the syndrome bits S.

**Error Locator** :to detect and locate errors by using ΔH and S.

**Error Corrector** : by ex-or operation of information bits and synbrome bits.

Decimal matrix code is also called as encoder reuse technique, as the decoder reuses the encoder which minimizes the area overhead.

## V. Result And Discussion

### *A. Router Input*

- data_in -> input data bus that transmits the packet from source network to another.
- read_enb_1 is active high input signal for reading the packet through output data bus data_out_1.
- pkt_valid -> active high input signal that detect an arrival of a new packet from sourc network.

**Fig. 4. Router input**

### *B. Router Output and Error Signal*

- val_out_X ->active high signal that detects that a valid byte is available for destination client network 2.
- data_out_X->output data bus that transmits data from the router to destination client network
- The data_out_X is fed as input to the the packet to corresponding router.
- Since no error will occur in simulation level, introduce of error signal is done to check the operation
- The 0th ,2nd bits of the dmc input signal d are inverted to get error signal u1.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 6, Issue 6, June 2019

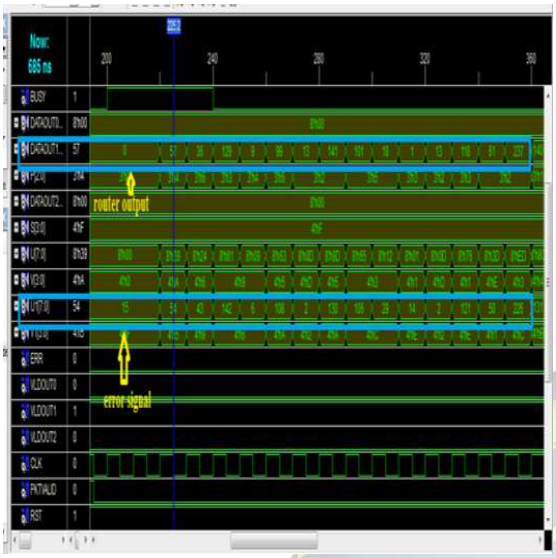


**Fig. 5. Router output and Error signal**

### C. DMC Corrected Output

- Using dmc error detection and correction technique the error is corrected.
- The correct signal dcorrect is obtained.

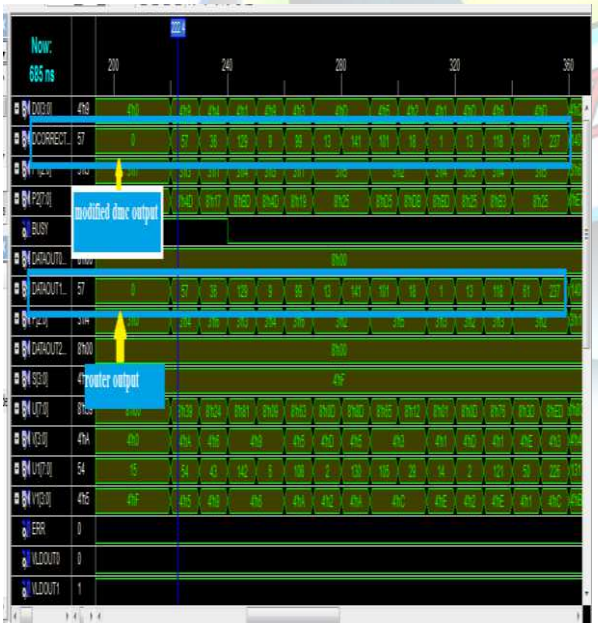


**Fig. 6. DMC corrected output**

### D. Layout diagram of overalltop module

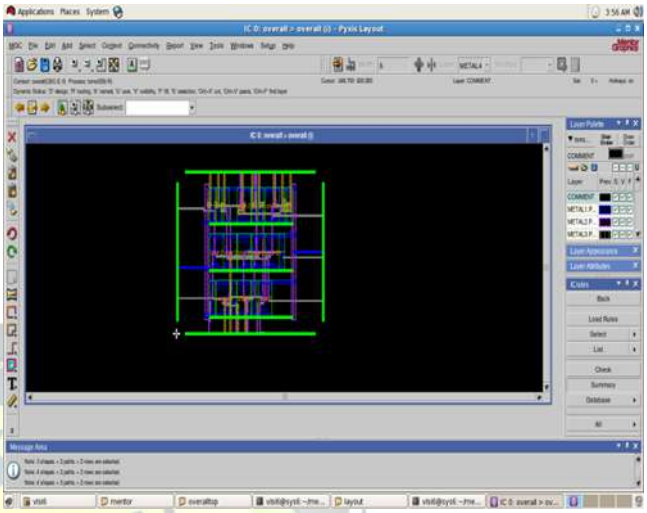

**Fig. 7. Layout diagram of overalltop module**

## VI. CONCLUSION

The proposed router becomes much reliable with the help of introducing decimal matrix code as error detection and correction technique. XY routing algorithm minimizes the deadlock situations. For 32-bit information the modified DMC requires 28 redundant bits while the Modified Decimal Matrix Code requires 36 bits. As Modified Decimal Matrix Code is also Encoder Reuse Technique (ERT) area overhead gets reduces. The simulation results for both router and MDMC are obtained. An attempt to reduce the number of redundant bits by using modernistic error correction and detection technique can be considered as a topic for future research.

## REFERENCES

[1]. Anil Kumar Singh(2016) , 'Error detection and correction by hamming code', International Conference on Global Trends in Signal Processing, Information Computing and Communication, pp. 35-37.

[2]. Anuja Naik , Tirumale .and K. Ramesh(2016) , 'Efficient Network on Chip (NoC) using heterogeneous circuit switched routers', International Conference on VLSI Systems,Architectures,Technology and Applications, pp. 1-6.

[3]. Lei Guo, Xiaorui Wang ,Yejun Liu , Pengchao Han ,and Yamin Xie (2017) 'Directional routing algorithm for deep space optical network', China ommunications,Vol.14, No.1,pp.158-168.

[4]. Liang Wang , Xiaohang Wang , and Terrence Mak(2016) , 'Adaptive Routing Algorithms for Lifetime Reliability Optimization in Network-on-Chip',IEEE Trannsactions on Computers, Vol. 65, No. 9, pp. 2896-2902.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 6, Issue 6, June 2019

[5]. Luis-J. Saiz-Adalid ,Pedro Reviriego , Pedro Gil , Salvatore Pontarelli , and Juan Antonio Maestro (2015) ,' MCU Tolerance in SRAMs Through Low-Redundancy Triple Adjacent Error Correction', IEEE Transactions on VLSI systems, Vol. 23, No. 10, pp. 2332-2336.

[6]. Martin Dario Arango Serna , Julian Andres Zapata Cortes , and Daniela Gutierrez Sepulveda(2015),'Modeling The Inventory Routing Problem (IRP) With Multiple Depots With Genetic Algorithms', IEEE Latin America Transactions, Vol.13, No. 12, pp. 3959-3965.

[7]. Rabab Ezz-Eldin , Magdy.A. El-Moursy , and Hesham F. A. Hamed (2016) 'Process Variation Delay and Congestion Aware Routing Algorithm for Asynchronous NoC Design', IEEE Transactions on VLSI systems, Vol. 24, No. 3, pp. 909-919.

[8]. Roshna Louis , Vinodhini. M. , and Murty. N.S(2015) , 'Reliable router architecture with elastic buffer for NoC architecture', International Conference on VLSI Systems,Architectures,Technology and Applications, pp. 1-4.

[9]. Timothy Girry Kale , Satoshi Ohzahata , Celimuge Wu ,and Toshihiko Kato(2016), ' Improving the tor traffic distribution with circuit switching method', IEEE 17th International Conference on High Performance Switching and Routing, pp. 106-107.

[10]. Geethu Jayan, Pavitha P. P. (2016), ' FPGA Implementation of an Efficient Router Architecture Based on DMC', International Conference on Emerging Technological Trends [ICETT], pp.1-6.