DESIGN AND IMPLEMENTATION SMART RELIABLE NETWORK ON CHIP

Nivedita S Naragundakar¹, Prof. Arun Raj S.R²
¹M.Tech Student, Digital Communication and Networking, U.B.D.T.C.E Davanagere, Karnataka, India
²Assistant Professor, Department of Electronics and Communication Engineering, U.B.D.T.C.E Davanagere, Karnataka, India

Abstract - Increasing complexity and the reliability evolution of SoCs, MPSoCs are becoming more sensitive to phenomena that generate permanent, transient, or intermittent faults. These faults may generate data packet errors and might even crash the router. The project is concentrated on the design and develop architecture for intelligent independent reliable routers like reliable router RKT-switch for implementation on FPGA. The work is carried on a new reliable dynamic NoC. The proposed NoC is a mesh structure of routers able to detect routing errors for adaptive routing based on the XY algorithm. The enhanced work is done in the project is to investigate the shortfalls of the different methods and the present technology and to find out the advancements of new system and also to analyze the scope for performance enhancements with respect to present system. New architecture will be simulated, implemented on FPGA and tested to reduce the delay/power consumption and area on chip for the given system. Coded using HDL Verilog language analyzed by synthesis using Xilinx ISE v14.5 and verified Simulation using Modelsim.

Keyword: Architecture of the reliable router RKT-switch, ECC, XY Routing Algorithm, Hamming Codes.

I. INTRODUCTION

Network on Chip (NoC) is slowly being accepted as an important paradigm for implementing communication among various cores in a SoC. With the advances in integrated circuits (IC) manufacturing a constant attempt has been to design enormous amounts of networks on the same chip so as to accomplish more resourceful and optimized chips. Further an efficient routing algorithm can be useful to increase the efficiency of the networks embedded on the chips [1]. Fig 1 shows the conceptual view of a NOC where, each tile is composed of a resource (R) and a switch or router(S). The router is connected to the four neighboring tiles and its local resource via channels. Each channel consists of two directional point-to-point links between two routers or a router and a local resource.

Increasing complexity and the reliability evolution of SoCs, MPSoCs are becoming more sensitive to phenomena that generate permanent, transient, or intermittent faults. These faults may generate data packet errors, or may affect router behavior leading to data packet losses or permanent routing errors. A fault in a routing logic will often lead to packet routing errors and might even crash the router. The precise location of permanent faulty parts of the NoC must be determined, in order for them to be bypassed effectively by the adaptive routing algorithm.

By using adaptive routing algorithm, When a router receives a data packet, it compares its own address to the destination and source addresses. Then, the router checks its own position in the deterministic XY path of the NoC for the considered data packet. The router performing this checking is able to decide whether the switch from which the packet was received made a routing error or not.
according to the correct XY path. However, this technique has a major drawback; it is unable to handle the bypass of faulty nodes and unavailable regions.

II. DESIGN METHODOLOGY

![FPGA Data Flow Diagram]

Above flow graph shows the method or procedure of the project, and each block present in the flow graph is briefly explained below:

**Specification:** Here Spartan 3 tool will use to work on this project, it has the operating speed of 100MHZ on board, and it also has soft processor Micro Blaze. And I work on tools like Xilinx ISE 13.4 and also work on simulation software ModelSim6.3c.

**Block Diagram:** Design of intelligent independent reliable routers like reliable router RKT-switch. The proposed NoC is a mesh structure of routers able to detect routing errors for adaptive routing based on the XY algorithm.
Logic Design: In this project logic design contains 4 router architecture includes loopback mechanism, I/O Buffers, FSM, Routing error detection and logic and hamming Logic(ECC).

Verilog: To work on this project Verilog language will use for the implementation of Design of like reliable router RKT-switch architecture in Xilinx ISE 13.4.

Synthesis: After combining, testing has to do, i.e whether the program is working properly or not, if yes it will continue with next process else it has to rewrite or correct. After correction again test the program, if its successfully working then it will be implement.

Implementation: All the process till testing will implement in this step.

Physical Dumping: In this step implementation of the project will dump into the FPGA.

Physical Testing: Finally the project will be test in the FPGA kit.

III. ERROR CORRECTION AND DETECTION
The error detection and correction which consists of Hamming Priority Encoder, XY Routing algorithm, Decoder and Random arbiter.

3.1 Hamming code
Hamming’s development [Ham] is a very direct construction of a code that permits correcting single-bit errors. He assumes that the data to be transmitted consists of a certain number of information bits \( u \), and he adds to these a number of check bits \( p \) such that if a block is received that has at most one bit in error, then \( p \) identifies the bit that is in error (which may be one of the check bits). Specifically, in Hamming’s code \( p \) is interpreted as an integer which is 0 if no error occurred, and otherwise is the 1-originated index of the bit that is in error. Let \( k \) be the number of information bits, and \( m \) the number of check bits used. Because the \( m \) check bits must check themselves as well as the information bits, the value of \( p \), interpreted as an integer, must range from 0 to \( m+k \) which is \( m+k+1 \) distinct values.

IV. ARCHITECTURE OF THE RELIABLE ROUTER RKT-SWITCH

![Fig 4. Architecture of the reliable router RKT-switch](image-url)
The proposed architecture is RKT-NOC switch; it will overcome both dead and live locks. It will support to any network & supports to adaptive routing algorithm and xy routing algorithm. It consists of loopback module, Routing error detection, Routing logic, Finite state machine, control logic, centralised journal of data packet errors, ECC. RKT Switch it consists of control signal, and data in, input control signal as the input to RKT Switch and data out, output control are act as output.

4.1 XY Routing

XY routing is one of the type of Dimension order routing (DOR) which is a typically a minimal turn algorithm and is more suitable for networks using mesh or torus topology. XY routing algorithm routes packets first in x-direction (or horizontal direction) to the correct column and then in y-direction (or vertical direction) to the receiver. In XY routing the addresses of the routers are their xy-coordinates. One of the advantages of XY routing is that it never runs into deadlock or livelock.

4.2 Random Arbiter

The Random Arbiter plays a vital role on the router in order to take decision for servicing a packet. The packets are serviced randomly from different directions of the network without any packet stacking. In the proposed arbiter there are five independent requests req_0 to req_4 and these requests are input to the arbiter. The arbiter will process and generate the grants GNT_0 to GNT_4 (service) for incoming packet’s requests in a random order on the router.

4.3 Flow chart for Random Arbiter

Fig 4.3. Flow chart for Random Arbiter
4.4 Synthesis of one node

4.5 Encoder
Encoder which consists of six inputs and 48 bits. The encoder which is used to encode the data and encoder output can be obtained by the process SEC-DEC. In encoder if the data is 8 input bits will obtain 13 bit output data.

4.6 Decoder
The decoder which consists of as 4 input and 1 output. The decoder which performs the reverse function of encoder and the parity bit is introduced in the decoder to detect and to correct the error.
4.7 Equations
1. \( \text{Latency}_{\text{min}} = N_{\text{RKT}} \cdot \text{latency}_{\text{RTRmin}} + N_{\text{RKT}} \cdot 2 \)
2. \( \text{Throughput}_{\text{max}} = M_{\text{P}} \cdot n \cdot \text{FIRmax} \cdot f \)

V. RESULTS

5.1 Simulation result of ECC

In ECC the input is given and error is introduced and ECC is corrected by using the Hamming code. Result of one node output if clk and rst is zero.
5.3. NOC OF 4X4

![Diagram of NOC OF 4X4](image)

5.4 RTL Schematic

![RTL Schematic](image)

*Fig 5.4 RTL Schematic*
5.5 Simulation Result of 4x4

Fig 5.5 Simulation result of 4x4
5.5 Design Summary

Table 5.5 Design Summary (Estimated values)

<table>
<thead>
<tr>
<th>Slice Logic Utilization</th>
<th>Device Utilization Summary</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Slice Registers</td>
<td>18</td>
<td>207,360</td>
<td>1%</td>
<td></td>
</tr>
<tr>
<td>Number used as Flip Flops</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Slice LUTs</td>
<td>29</td>
<td>302,360</td>
<td>1%</td>
<td></td>
</tr>
<tr>
<td>Number used as logic</td>
<td>29</td>
<td>207,360</td>
<td>1%</td>
<td></td>
</tr>
<tr>
<td>Number using O6 output only</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of occupied Slice registers</td>
<td>28</td>
<td>51,640</td>
<td>1%</td>
<td></td>
</tr>
<tr>
<td>Number of LUT Flip Flop pairs used</td>
<td>47</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number with an unused Flip Flop</td>
<td>29</td>
<td>47</td>
<td>6.1%</td>
<td></td>
</tr>
<tr>
<td>Number with an unused LUT</td>
<td>18</td>
<td>47</td>
<td>30%</td>
<td></td>
</tr>
<tr>
<td>Number of fully used LUT/FF pairs</td>
<td>0</td>
<td>47</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>Number of unique control sets</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of slice register sites lost to control set restrictions</td>
<td>6</td>
<td>207,360</td>
<td>1%</td>
<td></td>
</tr>
<tr>
<td>Number of bonded LUTs</td>
<td>57</td>
<td>969</td>
<td>5%</td>
<td></td>
</tr>
<tr>
<td>IO6 Flip Flops</td>
<td>4</td>
<td>31</td>
<td>3%</td>
<td></td>
</tr>
<tr>
<td>Number of BLIFG/BLIFGCTRLs</td>
<td>1</td>
<td>32</td>
<td>3%</td>
<td></td>
</tr>
<tr>
<td>Number used as BLIFG</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average Fanout of non-Clock Nets</td>
<td>2.25</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.4 Timing Summary

- Minimum input arrival time before clock: 1.455ns
- Maximum output required time after clock: 4.985ns
- Maximum combinational path delay: 5.208ns

The result is obtained by considering the RKT Switch to process data packets. The table 5.4 which gives relation between the Number of slice register, Slice LTUs, Slice FFs and the corresponding frequency for different meshes such as 1x1, 2x2, 3x3, 4x4, 6x6. The number os slice registers increases as the Noc sizes increases from 1x1 to 6x6. LUT also increases from 145 to 6113. For 4x4 Noc requires 4971 registers, 8826 LTUs and it can operate up to 92.885 MHz. It is smart Noc it correct the errors and run without failure.

<table>
<thead>
<tr>
<th>Noc</th>
<th>Slice Registers</th>
<th>Slice LUTs</th>
<th>Slice FFs</th>
<th>f[MHz]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1x1</td>
<td>138</td>
<td>145</td>
<td>124</td>
<td>495.368</td>
</tr>
<tr>
<td>2x2</td>
<td>379</td>
<td>456</td>
<td>343</td>
<td>457.143</td>
</tr>
<tr>
<td>3x3</td>
<td>956</td>
<td>1197</td>
<td>866</td>
<td>466.745</td>
</tr>
<tr>
<td>4x4</td>
<td>4971</td>
<td>8826</td>
<td>5273</td>
<td>92.885</td>
</tr>
<tr>
<td>6x6</td>
<td>4746</td>
<td>6113</td>
<td>4354</td>
<td>457.047</td>
</tr>
</tbody>
</table>

Table 5.4. Rkt-Noc Synthesis Results

5.5 Applications
- SMART antennas
- On chip network topologies
- Wireless LAN

VI. CONCLUSION

In this project, here I proposed new reliable router RKT Switch, Which is 4X4 mesh network topology. We are using for error detecting and correcting hamming codes, which is single bit correction technique. We are using for routing Adaptive Modified XY routing algorithm, where the main difficulty is to distinguish the bypasses of an unavailable component in the NOC (due to the use of the adaptive algorithm) from Real routing errors. Here I am not using any separate loopback module to process data or get back the data. The proposed design optimized in terms of area (LUT’s and Slices) and works at high speed.
Future work

In future, I focus on evaluating accurately the impact of faulty detection blocks and improving the routing error detection mechanisms, by protecting the DAI links and routing detection blocks against errors.

REFERENCES


