International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 7 Issue 1 –MARCH 2014.

COLLISION FREE PERMUTATION ON-CHIP NETWORK BY USING VHDL FOR SoC APPLICATIONS

K.K.Sasikala
PG Scholar
EEE Department
K.S.R College of Engineering
Tiruchengode

B.Yuvarani
Assistant Professor
EEE Department
K.S.R College of Engineering
Tiruchengode

ABSTRACT
The increasing complexity of integrated circuits drives the research of new intra-chip interconnection structures. A network-on-chip concept which is used in the distributed systems and computer networks subject areas to connect Intellectual Property cores in a structured and scalable way, this paper presents the Network-on-chip is used to support guaranteed traffic permutation in multiprocessor system-on-chip. The proposed system employs a circuit-switching technique and also it combined with a dynamic path-setup initiative under a Clos network. The dynamic path-setup method enables runtime path arrangement for arbitrary traffic permutations. A self-sufficient adaptive system for detecting and bypassing permanent errors in on-chip interconnects. The proposed structure reroutes data on faulty links to a set of spare wires without interrupting the data flow. To detect constant errors at runtime, a In-Line Test (ILT) method and a test pattern generator is used and also an improved syndrome storing-based error detection (SSD) method is used. Each detection method (ILT and SSD) is integrated independently into the non-interrupting adaptive system.

Keywords
Multistage Interconnection Network, Network-on-Chip, Permutation Network, Circuit-Switching, Traffic Permutation.

1. INTRODUCTION

A development of Multi-Processor System-on-Chip (MPSoC) design being interconnected with Network-on-Chip is currently emerging technology for applications of parallel processing, scientific computing, Speech Signal Processing, Image and Video Signal Processing and Information Technologies [1]–[4]. The main reason behind in this the implementation of this project is on-chip network employs based on a circuit-switching technique with a dynamic path-setup method under a multistage or CLOS network topology. The dynamic path setup tackles the analysis of runtime path arrangement for conflict-free permuted data. A three stage Clos network is a family of multistage networks is applied to manufacture scalable commercial multiprocessors with thousands of nodes in macro systems [6], [10]. This network has a rearrangeable property [10] that can recognize all possible permutations between its input and outputs. The option of the three-stage Clos network with a submissive number of middle-stage switches is used to minimize the implementation cost, whereas it still have a rearrange able Property for the network.

Circuit - switching technique is designed for use with the proposed network. This technique has three phases: the setup, the transfer, and the release [2], [8]. Packet switching needs an excessive amount of on-chip power and area for the queuing buffers (FIFOs) at the switching nodes and/or network interfaces [3]–[5].A dynamic path-setup system is the proposed method to support a runtime path arrangement when the permutation is changed. Every path setup, which starts from an input to find a path leading to its resultant output, is based on a dynamic probing mechanism. The concept of probing is introduced in works [2], [8], in which a probe (or setup flit) is dynamically sent under a routing algorithm in order to establish a path towards the destination. Exhausted profitable backtracking (EPB) [11] is proposed to use to route the probe in the network. A path array with full permutation consists of sixteen path setups, whereas a path deal with partial permutation may consist of a subset of sixteen path setups.

The project implements the circuit-switching technique with a dynamic path-setup scheme under a multistage network or CLOS network topology. The Circuit-switching technique has three phases: the setup, the transfer, and the release. A dynamic path-setup scheme supporting the runtime path arrangement occurs in the setup phase. Sequentially to support this circuit-switching method, a switch-by-switch interconnection with its handshake signals is proposed. The dynamic path setup is used to select the path routinely in the network while that path processing other data which deal with the challenge

of runtime path arrangement for conflict-free permuted data. CLOS network topology is a multistage network topology which is used in switching technique for data transfer in three stages, and also it has sixteen inputs and outputs, each path is selects dynamically according to the input given.The main advantage of network is that connection between a large number of input and output ports can be made by using only small-sized switches.

This will be an effective method to eradicate the problem of Packet-Switching technique such as an excessive amount of on-chip power and area for the queuing buffers, with pre-computed queuing depth at the switching nodes. It is mainly used in Speech Signal Processing, Image and Video Signal Processing and Information Technologies, PC interface (USB, PCI, PCI-Express, IDE) Computer peripheries (printer control, LCD monitor controller, DVD controller).Data Communication like wire line Communication: 10/100 Based-T, XDSL, Gigabit Ethernet, Wireless communication: Bluetooth, WLAN, 2G/3G/4G, WiMax. NoC technology applies networking theory and methods to on-chip communication and brings notable improvements over conventional bus and crossbar interconnections. NoC enhance the scalability of SoCs, and the power efficiency of complicated SOCs compared to other designs. NoCs have an inherent redundancy that helps tolerate faults and deal with communication bottlenecks.

2. SYSTEM DETAILS

The system design includes software design and hardware implementation, on-Chip network topology, Dynamic path-setup scheme.

2.1 System-on-Chip

System-on-chip is an integrated circuit (IC) that integrates all components of a computer or other electronic system into a single chip. It contain digital, analog, mixed-signal, and often radio-frequency functions every on a single chip substrate System on Chip on a VLSI chip that has all needed analog as well as digital circuits, processor and software. The demands on computer chips and processors these days are staggering. Even the simplest computer is required to complete complex tasks simultaneously. The benefits of SoC are self-evident everything needed to run the computer is contained in that one chip-the smaller and better. This includes the computer's operating system, electronic functions, memory of all variety, timers, interfaces like USB and Fire-wire, voltage regulators, timers, microprocessors, and basic function software applications. The chip has all that is required to run still detailed computer functions.

SOC = Chip + Software + Integration

2.2. Clos Network

Clos network is a multistage switching network. The advantage of network is that connection between a large number of input and output ports can be made by using only small-sized switches. In the figure n represents the number of sources which feed into each of the m ingress stage crossbar switches. There is exactly one connection between each ingress stage switch and each middle stage switch. And each middle stage switch is connected exactly once to each egress stage switch. Clos network is non-blocking when m ≥ n-1, Ingress/egress stage has r n×m switches, Middle stage has m r×r switches, and each switch at ingress/egress stage connects to all m middle switches.

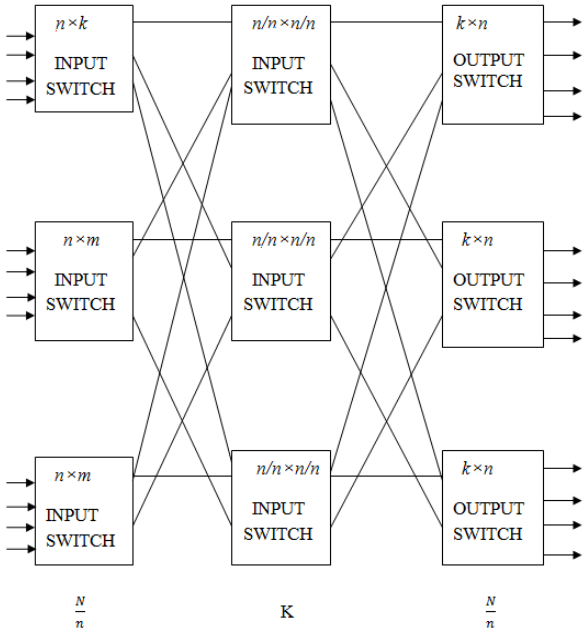

Fig 1: 3-Stage CLOS network

Figure 1 shows 3-stage Clos network, this network has a rearrangeable property that can realize all possible permutations between its input and outputs. The variety of the three-stage Clos network with a modest number of middle-stage switches is to minimize implementation cost, whereas it still has a rearrange able property for the network. A dynamic path-setup scheme supporting the runtime path arrangement occurs in the setup phase. Sequentially to support this circuit-switching method, a switch-by-switch interconnection with its handshake signals is introduced.

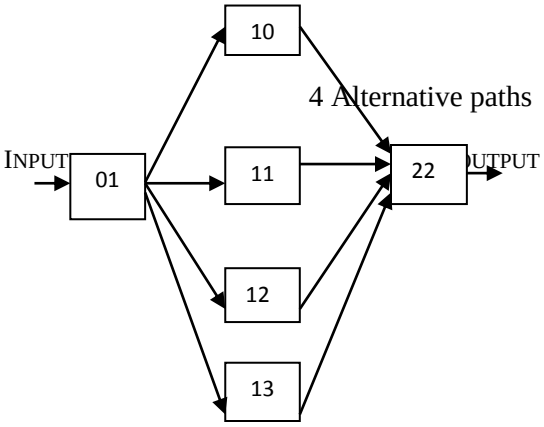

Fig 2: Switch-by-Switch Interconnections

Switch-by-Switch Interconnections, as shown in figure 2 , Phi-Hung Pham et al[20] .The bit format of the handshake include a 1-bit Request (Req) and a 2-bit Answer (Ans) Req = 1 is used when a switch requests an idle connection leading to the corresponding downstream switch in the setup phase. The Req = 1is also kept during data transfer along the set up path. A Req=0 denotes that the switch releases the occupied connection. This code is also used in together the setup and the release phases. An ans = 01(Ack) means that the destination is ready to receive data from the source. When the ans = 01 propagates back to the source, it denotes that the path is set up, then a data transfer can be usual instantaneously. An ans = 11(nAck) is reserved for end-to-end flow control when the receiving circuit is not ready to receive data due to being busy with other tasks, or swell out at the receiving buffer, etc. An ans = 10(Back) means that the link is blocked.

2.3. Network-on-Chip

Network on Chip is a communication subsystem on an integrated circuit, typically between IP cores in a System on a Chip (SoC). NoCs can span synchronous and asynchronous clock domains or use unclocked asynchronous logic. NoC tools apply networking theory and methods to on-chip communication and bring remarkable improvements over conventional bus and crossbar interconnections. NoC improves the scalability of SoCs, and the power efficiency of complicated SOCs compared to other designs. Network on chip is an emerging for communications within large VLSI systems implemented on a single silicon chip. As the number of IP modules in Systems-on-Chip (SoCs) increases, bus-based interconnection architectures may prevent these systems to meet the performance required by many applications. For systems with demanding parallel communication requirements buses may not provide the required bandwidth, latency, and power consumption. A solution for such a communication bottleneck is the use of an embedded switching network, called Network-on-Chip (NoC), to interconnect the IP modules in SoCs. In addition, NoCs have an inherent redundancy that helps tolerate faults and deal with communication bottlenecks. This enables the SoC designer to find appropriate solutions for different system characteristics and constraints.

3. CIRCUIT SWITCHING

Circuit switching is implementing a telecommunications network in which two network nodes establish a dedicated communications channel (circuit) through the network before the nodes can communicate. The circuit guarantees the full bandwidth of the channel and remains connected for the duration of the communication session. The circuit functions as if the nodes were physically connected as with an electrical circuit. This scheme has three phases: the setup, the transfer, and the release. In modern circuit-switched networks, electronic signals bypass through several switches before a connection is established. The resources remain dedicated to the circuit during the entire data transfer and the entire message follows the same path. Circuit switching can be analog or digital. With the prolonged use of the internet for voice and video, analyst envisages a gradual shift away from circuit-switched networks. A circuit-switched network is excellent for data that needs a constant link from end-to-end, for example, real-time video. Three kinds of switches are designed for the proposed on-chip network.

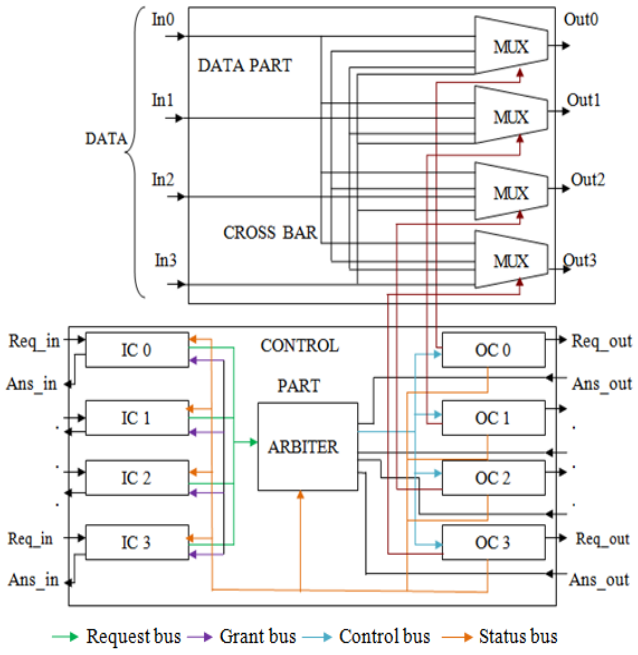

Fig 3: Common Switch Architecture

As Figure 3 shows Common Switch Architecture, Phi-Hung Pham et al[20] . it has common architecture has basic components: Input Controls (ICs), Output Controls (OCs), Arbiter, and a Crossbar. Incoming probes in the setup phase can be transported through the data paths to save on wiring costs. The Arbiter has two functions: first, cross-connecting the Ans_Outs and the ICs through the Grant bus, and second, as a referee for the requests from the ICs. When an incoming probe arrives at an input, the corresponding IC observes the output status through the Status bus, and requests the Arbiter to grant it access to the corresponding OC through the Request bus. When accepting this request, the Arbiter cross-connects the corresponding Ans_Out with the IC through the Grant bus with its first function. With the second function, the Arbiter, based on a pre-defined precedence rule, resolves contention when several ICs request the same free output. After this resolution, only one IC is established, whereas the rest are answered as facing a blocked link (i.e., similar to receiving an Ans=Back). Back code is used for a backpressure flow control of the dynamic path-setup Scheme.

on chip network topology with port addressing scheme as shown in Figure 4, Phi-Hung Pham et al[20] .in this network topology has 16-bit data with dynamic path-setup scheme is the key point of the proposed design to support a runtime path arrangement when the permutation is changed. Each path system, which starts from an input to find a path leading to its corresponding output, is based on a self-motivated probing mechanism.

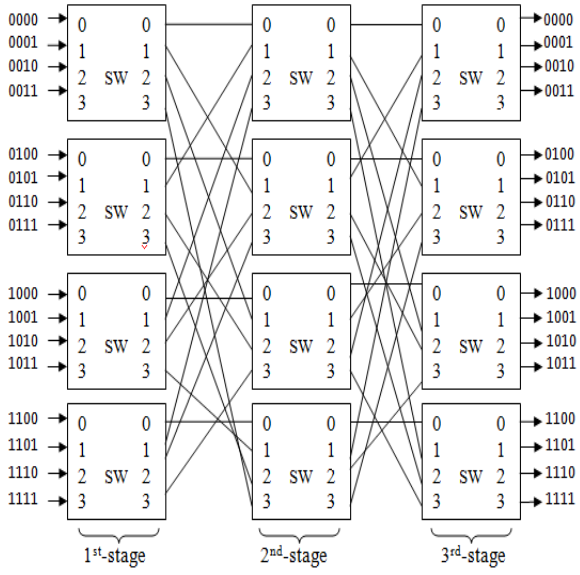

Fig 4: on chip network topology with port addressing scheme

A path arrangement with full permutation consists of sixteen path setups, whereas a path arrangement with incomplete permutation may consist of a subset of sixteen path setups. But in this method the error may be occur during transmission of bits, to avoid this bit error two methods are used namely as, In-Line Test method and Syndrome storing based error detection method (SSD) .

3.1 Spare Resources for Handling Permanent Errors

The use of spare modules to replace erroneous ones, particularly in array structures, is a long-known fault-tolerant approach [13]. Spare cells and wires have been used in field-programmable gate arrays to bypass defective components [14], [15]. Refan et al use spare wires to recover from switch failure by connecting each processing element to two switches in a network-on-chip (NoC) [16]. If a permanent fault occurs in one switch, processing elements share the working switch, and the system reroutes its data accordingly. Grecu et al have analysed the use of spare wires in NoCs [17] to increase manufacturing yield; reconfiguration of the links used crossbar switches with redundant channels. Unfortunately, the authors did not discuss the error detection procedure. In another work, Grecu et al. presented a built-in self-test methodology for NoC interconnects [18] and thoroughly discussed manufacture testing methods for NoCs, but they did not address runtime failures. Reick et al discuss dynamic I/O bitline repair using spare wires [19], but their detection and correction processes are not specified. Runtime reconfiguration has been presented in the authors' previous work [9], in which spare wires and the SSD method were applied to a self-timed system using ARQ. That prior work had a few significant limitations; for example, the system could tolerate only one permanent error per code interleaving section, and the data flow had to be stopped for the reconfiguration. In contrast, the ILT method proposed in this paper handles as many permanent errors as there are spare wires. Furthermore, It allows the link to be reconfigured without interrupting data transmission. This paper also presents an improved noninterrupting implementation of the SSD method.

Permanent-error correction in on-chip links using spare wires is a two-step process. First, the permanent error must be detected; then, the link must be reconfigured to avoid transmitting over the faulty wire. The proposed Reconfigurable adaptive link system framework is shown in Figure 5. It consists of a transmitter, a link, and a receiver. The incoming -bit-wide data word is encoded in the transmitter to a codeword of width , which is transmitted through the link and decoded in the receiver. The decoder is responsible for correcting any errors and outputs the original –bit data word. A number of spare wires are available. Reconfiguration units at the transmitter and

receiver determine which of the n + s lines carry data and which are left idle.

The reconfiguration control units pass reconfiguration information between the receiver and transmitter and synchronize reconfiguration. The error detection and reconfiguration central control unit detects permanent errors and initiates reconfiguration. The inputs to this unit depend on which detection method is used. For the SSD method, the syndrome and error vector from the decoder are needed. For the ILT method, test outputs (test_out) from the spare wires under test are needed. The ILT method requires a test pattern generator (TPG) block and test inputs (test_in) to produce test signals. We apply our techniques to permanent and intermittent errors in the link; the logic units are assumed to function correctly. Two methods are presented here, namely, the proposed ILT method and the improved SSD method.

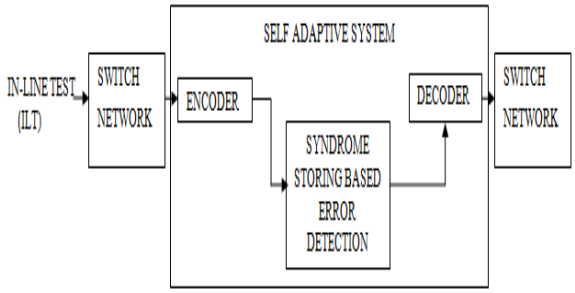

Fig 5: Reconfigurable link adaptive system

3.2 ILT Method

The proposed ILT method sequentially routes data from each pair of adjoining wires to a set of accessible spare wires, allowing each pair to be tested for intermittent and permanent faults. This is achieved during normal operation, without interrupting data transmission. To protect against runtime permanent errors, the ILT is run periodically, with a period that can be shortened to improve error resilience or increased for energy efficiency. In addition to this periodic testing, the ILT can be triggered when an error is detected beyond the error correction capability of the code protecting the link. The trigger can use a simple timer or be adaptively controlled by an upper protocol layer (e.g., application) to save energy during idle periods.

3.3 ILT Procedure

To begin the test, the ILT control unit reconfigures the link such that the first pair of wires is connected to the TPG (the data on those lines are rerouted to spare wires). The TPG issues a series of test patterns using test_in the signal. The ILT control unit compares the received test_out signal to a lookup table to determine if there is a permanent error in that pair of wires. The lookup table indicates which line(s) need(s) to be flagged as erroneous. Functional wires are reconfigured to carry data once again, and the process is repeated for each pair of wires. To provide even greater protection against permanent errors, the system is designed to take into account the number of spare wires when running a round of tests. The ILT control unit determines the number of remaining spares by checking the status of the last wire in the link .If one spare remains, the upper layer system can be alerted to the lack of spare resources, and the system only reroutes one wire at a time to that remaining spare. Instead of the two-wire test, the system performs a single-wire test that can detect opens in the line but cannot detect a short between the wire under test and its neighbors. If a wire adjacent to the wire under test is disabled due to a previously detected error, the ILT unit will perform the two-wire test on that pair. If no spare wires remain, the system will periodically retest each disabled wire in an effort to recover from intermittent errors.

3.4 SSD Method

Syndrome decoding is a common technique for decoding linear block codes. The syndrome is calculated by matrix multiplication $S = UH^T$, where U is a received code word vector of length $n(u = c + e)$ where c is a transmitted code word and is an error vector, both of length n), H^T is the transpose of $(n - k) \times n$ the parity-check matrix, and is a syndrome vector of length n-k. The syndrome gives the minimum weight error vector index, so the error vector e can easily be determined. The correction is done by $c = u + e$, eliminating the error from the received data word. The basic idea behind SSD is that the error syndrome of an error control code contains information about the errors of a received code word. If the syndromes of a number of consecutive received code words are the same, then it can be concluded that there is a permanent error in the link (limitations described momentarily). The error location can be extracted from the syndrome using the normal decoding procedure. The effectiveness of this approach comes from the fact that it takes advantage of the code and decoder already present at the system [12].

If there are more errors in the link than the error correction code is capable of correcting, the syndrome will be decoded incorrectly, providing a wrong error location. An important design decision for the SSD method is to determine how many syndromes to consider before deciding that an error is permanent In SSD, there is no method for recovering spare wires once they have been assigned, so the

error observation period can result in wasted wire resources if set too short. On the other hand, too long an observation period may result in a large number of cycles before the error is detected or may even leave errors undetected. This is because the detection of stuck-at faults is data dependent in order to be detected, the error must occur in all data words during the observation period.

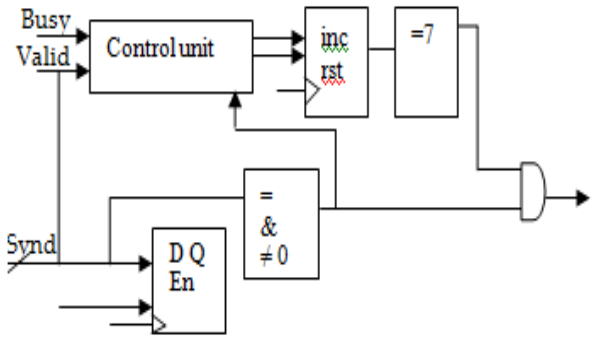

Fig 6: General Structure of SSD

The structure of the SSD circuit is shown in Figure 6. It contains a register to store the syndrome, a comparator between the input and the last syndrome, and a counter for counting the number of identical but nonzero syndromes. The counter is reset during a reconfiguration procedure, which is achieved using the busy signal. The counter only counts to seven since only eight comparisons are necessary. The first comparison is between the first and the second data sample. An error to the output is signaled when the seven most recent comparisons have been equal and the eighth comparison is also equal, thus resulting in an observation period of nine.

4. RESULTS AND DISCUSSION

The Network-on-Chip implementation consists of hardware and software part. The software part consists of the simulation of the hardware implementation. The simulation tool is Modelsim SE, The version uses in this simulator 6.3f. By using VHDL coding the output waveform is obtained. Switches in on-chip-network, 16-bit data is processing and get output waveform. It has permutation dynamic path set-up arrangement, so the data path will be change dynamically. clock pulse is given and reset set initially set as '1' and after set to '0' value to process the 16-bit data. When reset value is '1' the output control gets '0' value, when it is turned into '1' it will get 4-bit value. By using, In-Line test and Syndrome Storing-Based error detection method, the errors are detected and the data is transmitting without collision in the System-on-Chip.

4.1 OUTPUT STAGES OF SWITCHES IN NETWORK

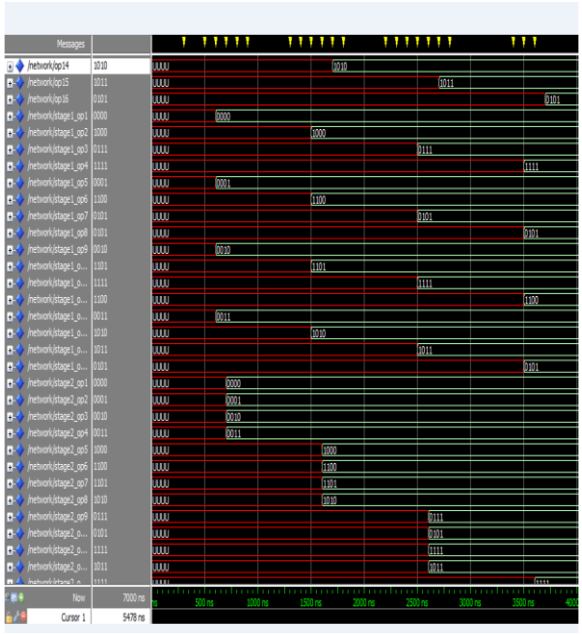

Fig 7(a) : 16 bit data output waveform

In three stages of switches in on-chip-network, 16-bit data is processing and get output waveform. It has permutation dynamic path set-up arrangement, so the data path will be change dynamically.

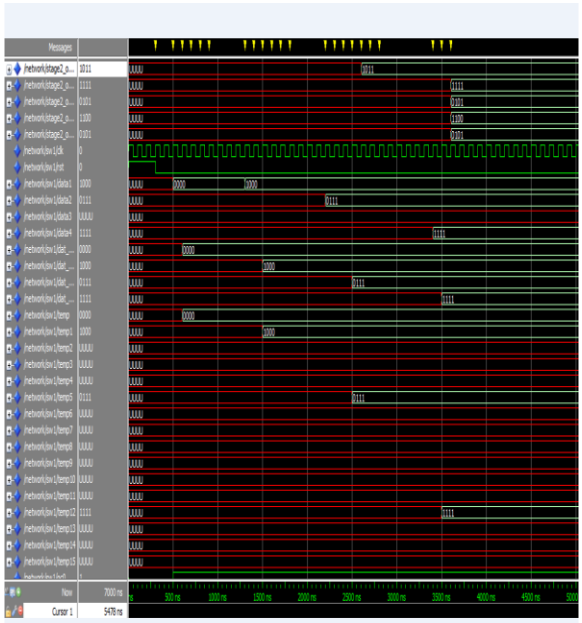

Fig 7(b) : 16 bit data output waveform

4.2 Output Waveform for Syndrome Storing Based Error Detection

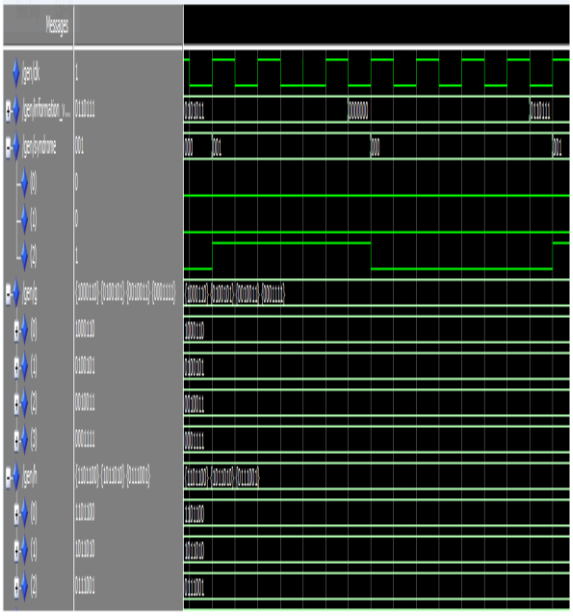

Fig 8: Syndrome generation waveform

Syndrome generation waveform is shown in Fig 8. The syndrome and also generator matrix is generated.

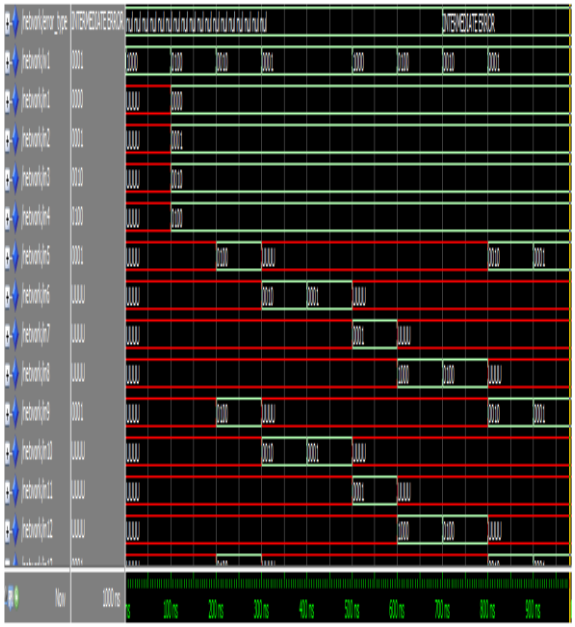

Fig 9: Intermediate Error waveform

Intermediate error waveform is shown in fig 9. While data processing in the switch network one bit change has been occurred. The error flagged as intermediate error. This error can be eliminating by invert operation.

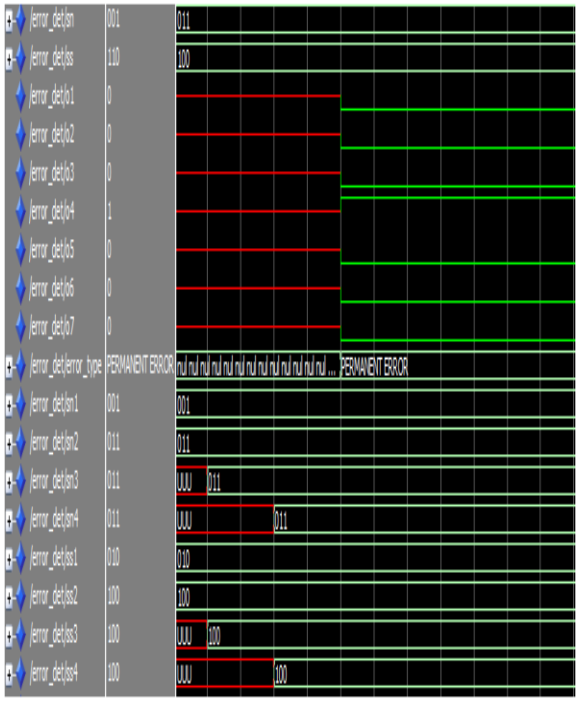

Fig 10: Permanent error waveform

Permanent error waveform is shown in Fig 10. While data processing in the switch network more than one bit change has been occurred and also continuously network received same error data that error flagged as permanent error. This error can be avoid by using Spare wires without interrupting another flow of data processing in the switch network.

5. CONCLUSIONS

Thus the method for the Network-on-Chip, the data processing in each switch by using Circuit-Switching technique with a dynamic path-setup scheme under a multistage network topology. The dynamic path setup is used to select the path routinely in the network while that path processing other data which deal with the challenge of runtime path arrangement for conflict-free permuted data. To detect permanent errors at runtime, a In-Line Test method using spare wires, Syndrome Storing-Based error detection method and a test pattern generator are used. By testing every wire in the link, the ILT method also recovers resources from intermittent errors that were incorrectly flagged as permanent. In addition to the ILT method, a number of important improvements to an alternative syndrome storing-based error detection method, which is based on evaluation of consecutive code syndromes at the receiver. Syndromes are calculated during the decoding procedure and contain information on errors in the received words.

6. ACKNOWLEDGMENT

I wish to express my deep sense of gratitude and thanks to all helped in my course of work.

7. REFERENCES

[1] Borkar, S. (2007) "Thousand core chips—A technology perspective", in Proceedings ACM/IEEE Design Automatic Conferences. (DAC), pp. 746–749.

[2] Pham, P.H., Mau, P. and Kim, C. (2009) "A 64-PE folded-torus intra-chip Communication fabric for guaranteed throughput in network-on-chip based applications", in Proceedings IEEE Custom Integrated Circuits Conferences (CICC), pp. 645–648.

[3] Neeb, C., Thul, M.J. and Wehn, N. (2005) "Network-on-chip-centric approach to interleaving in high throughput channel decoders", in proceedings IEEE International Symposium Circuits Systems (ISCAS), pp. 1766–1769.

[4] Moussa, H., Baghdadi, A.and. Jezequel, M. (2008) "Binary de Bruijn on-chip network for a flexible multiprocessor LDPC decoder", in Proceedings ACM/ IEEE Design Automatic Conferences. (DAC), pp. 429–434.

[5] Vangal, S.R., Howard, J., Ruhl, G., Dighe, S., Finan, D., A. Singh, T. Jacob, S. Jain, Erraguntla, Roberts, C., Hoskote, Y., Borkar, N. and Borkar, S.(2008) "An 80-tile sub-100-w Tera FLOPS processor in 65-nm CMOS", IEEE J. Solid-State Circuits, Vol. 43, no. 1, pp. 29–41.

[6] Dally W.J. and Towles, B. (2004) Principles and Practices of Interconnection Networks: San Francisco, CA: Morgan Kaufmann.

[7] Michael, N., Nikolov, M., Tang A., Suh, G.E. and Batten, C. (2011) "Analysis of application-aware on-chip routing under traffic uncertainty," in Proceedings IEEE/ACM Interational Symosium Network-on- Chips (NoCS), pp. 9–16.

[8] Pham, P.H., Park,J., Mau, P., and Kim, C. (2012) "Design and implementation of backtracking wave-pipeline switch to support guaranteed throughput in network-on-chip", IEEE transactions on very large scale integration (VLSI) systems.

[9] Ludovici, D., Gilabert, F., Medardoni, S., Gomez, C., Gomez, M.E., Lopez, P., Gaydadjiev, G.N. and Bertozzi, D. (2009) "Assessing fat-tree topologies for regular network-on-chip design under nanoscale technology constraints", in Proceedings Design Automatic Test Europe ,Conferences (DATE), pp. 562–565.

[10] Yang, Y. and Wang, J. (2004) "A fault-tolerant rearrangeable permutation network", IEEE Transactions on Computers., Vol. 53, no. 4, pp. 414–426.

[11] Gaughan, P.T. and Yalamanchili, S. (1995) "A family of fault-tolerant routing Protocols for direct multiprocessor networks", IEEE transactions on Parallel Distribution Sysemts., Vol. 6, no. 5, pp. 482–497.

[12] Lehtonen, T., Liljeberg,P. and Plosila, J. (2007) "Online reconfigurable self timed links for fault tolerant NoC", VLSI Designation., Vol. 2007, pp.1–13.

[13] Johnson,B.(1989) " Design and Analysis of Fault Tolerant Digital Systems", Boston, MA: Addison-Wesley.

[14]Hanchek, F. and S. Dutt, S. (1998) "Methodologies for tolerating cell and interconnect faults in FPGAs", IEEE Transcations on Computers", Vol. 47, no. 1, pp.15–33.

[15] Yu, and G. G. F. Lemieux, G.G.F (2005) "Defect-tolerant FPGA switch block and connection block with fine-grain redundancy for yield enhancement," in Proc. International Conferences in. Field Programming Logic", pp. 255–262.

[16] F. Refan, F., H. Alemzadeh, H., S. Safari, S., P. Prinetto, P. and Z. Navabi, Z. (2008) "Reliability in application specific mesh-based NoC architectures", in Proceedings IEEE IOLTS, pp. 207–212.

[17] Grecu, C., A. Ivanov, A., R. Saleh, and P. P. Pande, (2006) "NoC interconnect yield improvement using crosspoint redundancy," in Proceedings 21st IEEE International Symposium DFT, pp. 457–465.

[18] C. Grecu, C., A. Ivanov, A., R. Saleh, R. and P. P. Pande, P.P. (2007) "Testing network-onchip communication fabrics", IEEE Transcations. Computer-Aided Design Integrated Circuits Systems, Vol. 26, no. 12, pp. 2201–2214.

[19] Reick, K., Sanda, P.N., Swaney, S., Kellington, J.W., Mack, M.J., Floyd, M.S and Henderson, D. (2008) "Fault-tolerant design of the IBM Power6 microprocessor," IEEE Microprocessor Vol. 28, no. 2, pp. 30–38.

[20] Phi-Hung Pham, Junyoung Song, Jongsun Park, and Chulwoo Kim. (2013) "Design and Implementation of an On-Chip Permutation Network for Multiprocessor System-on-Chip", IEEE transactions on very large scale integration (VLSI) systems, Vol. 21, No. 1, pp. 234-239.