An Efficient Fault Tolerant Routing Interconnect System for Neural NOC

Dr.A.Pradeep kumar[1], Y. Devendar Reddy [2], Dr.T.Srinivas Reddy [3], K. Jamal [4]
{pradeepkumar@mrec.ac.in[1], devender_reddy03@rediffmail.com[2], srinivasreddy.14@mrec.ac.in[3], kjamal24@gmail.com[4]}

Professor, Mallareddy Engineering College (A), Hyderabad, Telangana, India[1], Associate Professor, Nalla Narasimha Reddy Education Society's Group of Institutions, Hyderabad, Telangana, India[2], Professor, Mallareddy Engineering College (A), Hyderabad, Telangana, India[3], Assistant Professor, GRIET(A), Hyderabad, Telangana, India[4]

Abstract. Large scale Neural Network (NN) accelerators typically have multiple processing nodes that can be implemented as a multi-core chip, and can be organized on a network of chips (noise) corresponding to neurons with heavy traffic. Portions of several NoC-based NN chip-to-chip interconnect networks are linked to further enhance overall nerve amplification capacity. Large volumes of multicast on-chip or cross-chip can further complicate the construction of a cross-link network and create a NN barrier of device capacity and resources. In this paper, this refer to inter-chip and inter-chip communication strategies known as neuron connection for NN accelerators. Interconnect for powerful fault-tolerant routing system neural NoC is implemented in this paper. Regarding intra-chip communication, this recommend crossbar arbitrage placement, virtual interrupt and path-based parallelization strategies for virtual channel routing, leading to higher NoC output with lower hardware costs. For multicast-based traffic. Regarding inter-chip communication, this propose a lightweight NoC compatible chip-to-chip interconnection scheme to allow efficient interconnection for NoC-based NN chips. In addition, this will test the proposed methods with four Field Programmable Gate Arrays (FPGAs) on four hardwired Deep Neural Network (DNN) chips. The experimental results show that the proposed interconnection network can effectively handle data traffic with high throughput and low DNN through advanced links.

Keywords: Chip-to-chip interconnection, Deep Neural Network (DNN), Hardware Accelerator, Interconnection Architecture, Network-On-chip (NoC).

1 Introduction

Technology can combine more and more logic circuits into a single chip. Therefore, the chips characters became more powerful. Processing units that work with different clock frequencies are integrated in a single chip [1] [2]. Traditionally, processing units have to interconnect the various sections of the SoC using bus structures.

However, bus SoCs are key connectivity schemes because their scalability is too low [1]. Network on a chip has been suggested as a possible candidate for reduced scalability and poor connectivity efficiency issues presented in the previous system on a chip. NoC uses network connectivity instead of bus systems to provide Globally Asynchronous and Locally

I3CAC 2021, June 07-08, Chennai, India

DOI 10.4108/eai.7-6-2021.2308772

Synchronous data transfer (GALS), which ensures that NoCs have increased the reliability of network connectivity and power usage by on-chip logic and device complexity [2].

The continuous increase in the size and complexity of the NoC interconnect infrastructure presents major challenges to the time-tested initiative of the architecture [3]. Chips up to 100 cores today rely on a wide range of NoC connectivity. In addition, active interconnecting architectures are being implemented to provide better efficiency and power utilization. This increasing maturity is obvious when one explores the growing concerts of features integrated into the router architecture, including complex arbitration processes, speculation and adaptive routing [4].

There is also the added difficulty of adopting advanced routing protocols, custom power and plug devices and dynamic overlay communication protocols. This high design complexity represents a large design area that cannot be fully implemented and verified during the design review phase. As a result, interconnect designs can be sold in some unconfirmed versions with hidden defects in the corners.

NoC is similar to the traditional embedded network used in parallel multiprocessor computers. However, unlike parallel computers, NoC has unique characteristics. The important variations between NoC and parallel computers are energy use restricted design specialization and the area and variety of materials used. Most of the applications used in battery-operated NoCs are locked in by power consumption. The low power consumption of the device is the main objective of the NoC structure.

In addition, NoC can be designed for separate and separate applications from parallel computer network for a variety of unknown applications. In addition to memory and processors, a variety of modules can be assigned to a single nose and include Digital Signal Processing (DSP) and Field Programmable Gate Array (FPGA) architecture, making noise more extensible. Applications implemented in NoC designs usually have limited configuration tools and strict performance characteristics. The main challenge for NoC architecture is how to meet application performance benchmarks with minimal capital. Capital constraints require more appropriate algorithms and more expensive NOCs [5].

As a result, the critical routing algorithm is now the primary routing algorithm for NoCs. With the level of NoC increasing, the growing debate over data transfer calls for expansion of the cache capacity of routers will increase. As a result, artificial channel models are installed in the router. Multiple data packets are stored in separate storage by setting physical storage partition to reduce router buffer load on chip. How to help reduce friction is an important way to improve router efficiency.

The buffer size on the NoC router needs to be reduced, which can be solved using virtual channels. The main challenge of the NoC architecture is how to meet application performance standards using minimal capital. Capital constraints require more appropriate algorithms and more expensive NOCs. As a result, the decisive routing algorithm has now become the primary routing algorithm for NoC. With the reduction of NoC, the growing debate about data transfer requires expanding the buffer capacity of routers.

As a result, synthetic channel models are installed in the router. Multiple data packets are stored in a separate memory by setting the physical memory partition to reduce the buffer load of the on-chip router. The rest of the data fits are tracked by the header flip in a pipeline manner. If the header blocks flight, the rest of the flight will also stop on simulation channels. Since packets need to be fixed in a single virtual channel instead of the entire buffer, routers can be configured with some default channel buffers for each terminal. Digital channels can also be used to reach higher channel capacity at lower cost.

2 Related Work

NN's hardware acceleration has attracted tremendous attention in recent years However, DNN-based interconnection networks have very few contributions. New- hub proposes a hybrid ring mesh for neuromorphic systems designed to accelerate the multilayer perception. In new-NoC, single-layer neurons are connected to a ring and single-ring neurons share the same data for multicast traffic. These local circuits are connected to each other through the NoC mesh to influence data movements between different layers. The ring topology is often affected by output and delay. Proposes a closed topology-based indirect interconnection network customized for close NN feed-forward NNs. This overcomes the narrow bending bandwidth of the tree and the large diameter of the mesh topology, which shows the reliability of the power supply in handling multicast traffic. However, the structure suffers from the physical limitations of the cable.

Eyeriss offers a Hierarchically Overlapping Network (HM-NoC) for DNNs. Processing Elements (PES) and Global Buffer (GLB) are grouped and linked by HM-NoC. NoC can be configured in several circuit-switched routing modes depending on the type of data transmitted. With enhanced NoC, different types of data (input activation, partial weights and volumes), from high bandwidth to high data reuse, can be exchanged between PEs and GLBs. The large tree topology was used for internal and network communication over Hyper Transport 2.0 to monitor data traffic between chips in large-scale NN architectures. However, advanced intra-chip and inter-chip architectures do not fully accept multicast DNN traffic, which can deter network amplification systems.

3 Literature Survey

Hierarchical agent architecture is suggested to provide on-line management capabilities for NoC-based applications. Unit structures are optimally maintained by agents at each construction level based on the circuit conditions monitored during runtime. This paper explores the monitoring relationship between the level of the agent and focuses on the alternatives for system optimization to be handled at various levels of the agent. They recommend the hierarchical configuration of the agent with the appropriate monitoring services. This architecture introduces a level of control into the NoC network hierarchy.

This layer provides the scalability and increased flexibility of large-scale NoC systems designed to maximize device performance by balancing all of the chip resources. Hierarchical approaches is used for multi-capacity management services and fault tolerance management.

The hierarchical agent simulation approach is excellent at achieving self-conscious and parallel computing in a scalable manner. A hierarchical agent that controls device status during runtime and re-configures components to boost performance in the event of an error.

The method of manufacturing a complex device, such as a chip network, will trigger several failures. Inexpensive routing algorithms are used in NoC to support permanent fault connections. Use appropriate simulation and synthesis to measure efficiency, power consumption and area overhead to see the effect of these algorithms. Proposed error-tolerant routing algorithms that can be reconstructed which make decisions based on local error information stored at each node and in the current and destination node configuration register which is obtained [6].

The first routing algorithm (FT XY) tolerates a fault link. (FT XY2) & (FT XY3) is an extension (FT XY) to find two other defective links, considering the hardware overhead gap. According to simulation and synthesis, the proposed routing algorithm does not enable VCs to have minimal overhead and overhead capability. The Fault-on-Neighbor (FoN) routing algorithm for NoC is proposed in clause, which sets out the routing decision on the basis of the connection status of the neighboring switches within 2 hops in order to prevent incorrect connections and switches. Diversion routing is a compatible routing algorithm that is essentially implemented to hardware, which ensures that packet buffers are not used while shipping. Fault – on - Neighbor (FoN) conscious variance routing algorithm based on the distribution of incorrect information in 2 zones to prevent defective links and switches and to preserve clear convex and concave error zones without blocking or blocking existence.

Fault-tolerant routing can be divided into two classes: random and critical. It transmits unwanted packets over various channels to stop random communication errors. The critical algorithm is used NoC architectural redundancy to transport packages to the destination through various means to achieve fault tolerance. The decision to route Forced Wormhole Routing (FWR) is based on the buffer state of the routing table and neighboring keys. Use first level packet as visibility to check queue and adjacent key buffer status. This section specifies the NoC fault-tolerant elastic routing algorithm based on the turn model.

When ensuring proper service, the switch can be rebuilt around broken components without the use of virtual networks. A fault-based positive deflection routing algorithm has been proposed that allows decision-making on cost-function-based routing. The switch implements an online troubleshooting approach and makes a routing decision based on a cost function that takes root duration and local fault status into account. Not only can it handle connecting and turning errors, but it can also handle crossbar errors. Since routing decisions are based solely on inaccurate knowledge of the current transfer, the packet hop count region can be easily overwhelmed by some faulty versions.

The geometry of the NoC is based on Nostrum NoC, a 2D mesh topology. It varies from a typical 2D network in that the limit output is attached to the same switch input and the packet sent in that direction is returned to the same switch. It can be used as a buffer for packets. Distraction routing is used to make a routing decision depending on the priority packet and the next network load voltage varies over the last 4 cycles. The two incoming packets are chosen on the basis of their hop count, which shows the number of hop packets. The packet with the largest number of hops shall have the highest preference. Requires high to low priority routing alternatives for packet relocation.

Device configurations are favorably controlled by agents at each construction level based on circuit conditions tracked at runtime. Device configurations provide periodically configured resource consumption and power supply. This technology provides a comprehensive approach to the design of VLSI (Very Large Scale Integration) circuits under the control of variations and strict power limits.

By implementing a bio-induced agent-based hierarchical modeling approach bio-induced system architecture, the design framework NoC proposed for hierarchical agent monitoring. Bio-induced approach-agent mechanisms have appealing network implementation characteristics such as scalability and compatibility, bio-induced system architecture divides conventional control resources across different agent layers and is more scalable than traditional systems.

They propose a hierarchical control factor based on an engineering approach, shaped by the joint efforts of hierarchical intelligent agents to plan, manage and fine-tune the NoC system at various operational levels, including system outputs, energy efficiency, fault tolerance and diversity. Provides high-level capture of concurrent control functions through distributed networks. Here, each stage of the agents carries out special controls on the basis of their detail. The monitoring process and tasks are distinguished by an unmistakable fixed structure of the system.

4 Fault Tolerant Routing Interconnect System For Neural NoC

The below figure (1) shows the architecture of fault tolerant routing interconnect system for neural NoC. A new routing strategy is being implemented to address imbalanced data-dependent network loads on multiple virtual networks. It has an eastern port, a western port, a northern port, a southern port and a domestic port. Each terminal has four simulation channels. An escape network can be set up by gently creating one channel for each router. The third default channel for each router is not accessible at the start of the transmission. The data packet is sent to the third virtual channel only when the queue time limit of other channels is reached and the other channels are sent to the destination without access.

The basic Y-X routing strategy is discarded in order to avoid competition at the exit ports. The leak network is used to pass packets from the usual channel to the leak channel only after data packets have been waiting for a long time in the simulation channel. However, data packets using X-Y routing in the normal channel and packets in the loss network must be transmitted simultaneously, since data packets in the normal channel and data packets in the loss channel which vary from the same output channel.

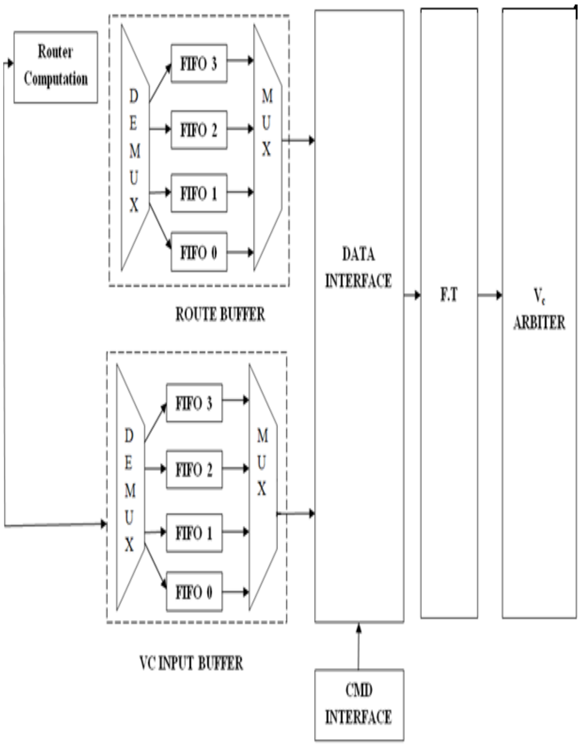

Fig. 1. Architecture Of Fault Tolerant Routing Interconnect System For Neural NoC

When NoC runs at a lower data injection rate, NoC's delay is greater than that of traditional mesh-based NoC. The condition is changing as the pace of data injection rises. This is because the data packets buffered in the virtual channel do not wait long when NoC has a lower data injection rate. The waiting time, however, does not reach the limit. It provides control and retrieval function for the reception and distribution of data to the entire arbitration system. Statements of VCs (Version Control systems) are mediated according to the credit of the target VC and the preference of the local VC.

In addition, it is responsible for sending and analyzing instructions. The credit synchronization system is used here to prevent overloading the RX handle. This module sends a request for a credit update based on the RX level and receives a credit order from another chip to change the local credit. If the credit is reduced to 0, the goal on the RX side means that the VC is complete.

For convenience, this has just mentioned five types of commands. When an error occurs, the data link layer sends a credit synchronization order only. In most instances, the machine sends good data instead of a command. Compared to PCI's (Personal Computer Interconnects) packet redundancy in the network connection layer, this architecture has fewer overheads and is likely to improve efficiency substantially.

5 Results

The below figure (2) shows the comparison of delay in fault tolerant routing interconnect system for neural NoC and routing interconnect system for neural NoC. From this figure it can observe that the fault tolerant routing interconnect system for neural NoC reduces the delay very effectively.

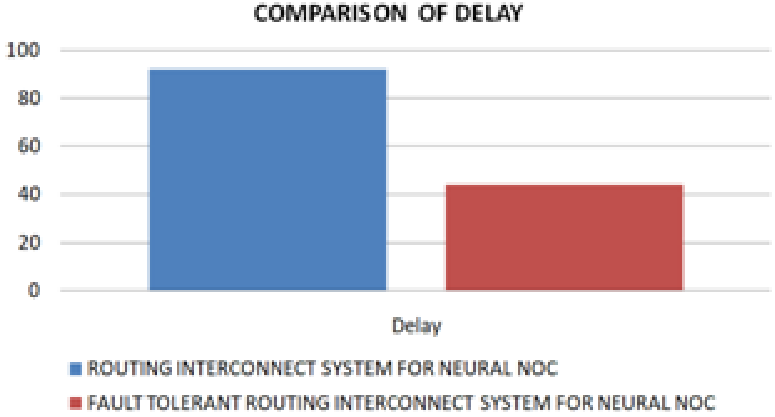

Fig. 2. Delay Comparison

The below figure (3) shows the comparison of accuracy of in fault tolerant routing interconnect system for neural NoC and routing interconnect system for neural NoC. In fault tolerant routing interconnect system for neural NoC accuracy is very high. It gives effective output and reduces the fault very efficiently.

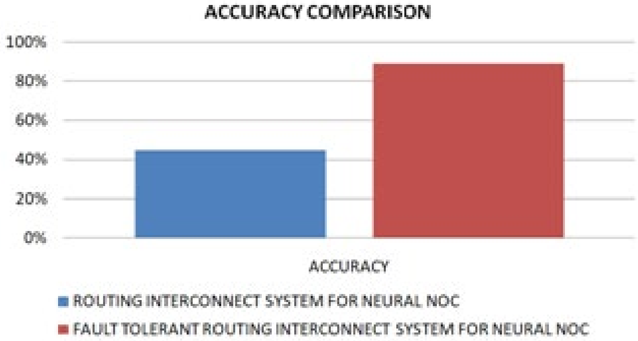

Fig. 3. Accuracy Comparison

The below figure (4) shows the comparison of reduction of number of faults in fault tolerant routing interconnect system for neural NoC and routing interconnect system for neural NoC. Compared to both number of faults are reduced in fault tolerant routing interconnect system for neural NoC.

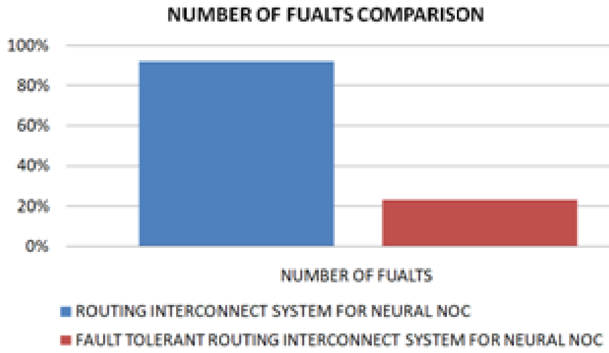

Fig. 4. Number of Faults

The below figure (5) shows the comparison of efficiency of fault tolerant routing interconnect system for neural NoC and routing interconnect system for neural NoC.

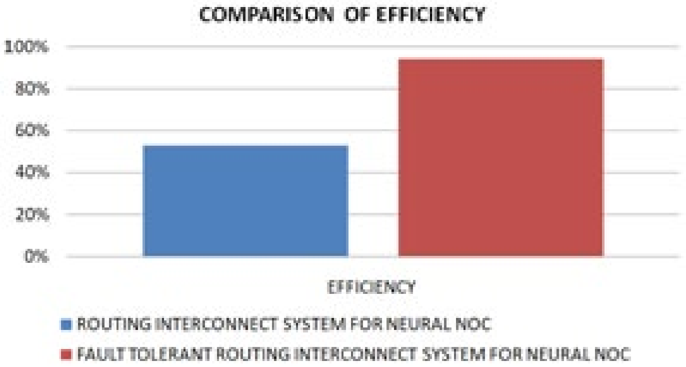

Fig. 5. Efficiency

6 Conclusion

Hence in this paper, this has suggested an efficient fault-tolerant interconnecting mechanism for neural NoC. Interconnection to handle the tremendous amount of multicast-based traffic in DNN accelerators effectively. The interconnection shall be thoroughly tested using the RTL (Register Transfer Level) time consistency model. It is also interconnected with four hardware systems focused on FPGA (Field Programmed Gate Array). From results it can observe that it occupies less area, reduces the delay and increases the accuracy of system.

References

[1] S. Yin et al., “An energy-efficient reconfigurable processor for binaryand ternary-weight neural networks with flexible data bit width,” IEEE J. Solid-State Circuits, vol. 54, no. 4, pp. 1120–1136, Apr. 2019.

[2] Y.-H. Chen, T.-J. Yang, J. S. Emer, and V. Sze, “Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices,” IEEE J. Emerg. Sel. Topics Circuits Syst., vol. 9, no. 2, pp. 292–308, Jun. 2019.

[3] K. Bhardwaj and S. M. Nowick, “A continuous-time replication strategy for efficient multicast in asynchronous NoCs,” IEEE Trans. Very Large Scale Integration. (VLSI) System, vol. 27, no. 2, pp. 350–363, Feb. 2019.

[4] M. F. Reza and P. Ampadu, “Energy-efficient and high-performance NoC architecture and mapping solution for deep neural networks,” in Proc. 13th IEEE/ACM Int. Symp. Netw.-Chip, Oct. 2019, pp. 1–8.

[5] X. Zhou et al., “Cambricon-S: Addressing irregularity in sparse neural networks through a cooperative software/hardware approach,” in Proc. 51st Annu. IEEE/ACM Int. Symp. Micro architecture (MICRO), Oct. 2018, pp. 15–28.

[6] Kwon, A. Samajdar, and T. Krishna, “MAERI: Enabling flexible dataflow mapping over DNN accelerators via reconfigurable interconnects,” in Proc. 23rd Int. Conf. Archit. Support Program. Lang. Oper. Syst. (ASPLOS), Mar. 2018, pp. 461–475.

[7] A. Firuzan, M. Modarressi, M. Daneshtalab, and M. Reshadi, “Reconfigurable network-on-chip for 3D neural network accelerators,” in Proc. 12th IEEE/ACM Int. Symp. Netw.-Chip (NOCS), Oct. 2018, pp. 1–8.

[8] B. Bohnenstiehl et al., “KiloCore: A 32-nm 1000-processor computational array,” IEEE J. Solid-State Circuits, vol. 52, no. 4, pp. 891–902, Apr. 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in Proc. Eur. Conf. Compute. Vis. (ECCV), Oct. 2016, pp. 630–645.

[10] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, “Optimizing FPGA-based accelerator design for deep convolution neural networks,” in Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays (FPGA), Feb. 2015, pp. 161–170.

[11] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, “Deep learning with limited numerical precision,” in Proc. Int. Conf. Mach. Learn. (ICML), Jul. 2015, pp. 1737–1746.

[12] A. Touzene, “On all-to-all broadcast in dense Gaussian network onchip,” IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 4, pp. 1085–1095, Apr. 2015.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580–587.

[14] P. Ou et al., “A 65 nm 39 GOPS/W 24-core processor with 11Tb/s/W packet-controlled circuit-switched double-layer network-on-chip and heterogeneous execution array,” in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2013, pp. 56–57.

[15] G. Hinton et al., “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” IEEE Signal Process. Mag., vol. 29, no. 6, pp. 82–97, Nov. 2012.