

Evolving Bio Plausible Design With Heterogeneous Noc

Nithya Narayanamurthy*, J. K. Periasamy *

*Department of Computer Science & Engineering, Sri Sairam Engineering College ,Tamil Nadu, India
nithya.murthi@gmail.com, jkperiasamy@gmail.com

Abstract— With the increasing complexity of various communication and multimedia standards, Network on Chip (NoC) is evolving as a solution for on-chip communication problems. Bio-inspired paradigms such as spiking neural networks (SNNs), when incorporated can take full advantage of their inherent parallelism and offer the potential to meet the demands of real-time fault tolerant applications. This paper presents a scalable, configurable 2-D irregular mesh network on chip (NoC)-based SNN architecture. The modules are placed in such a way to minimize the spatial traffic density, unnecessary switch links and nodes resulting in reduced SNN area requirements. The proposed design is based on a planar grid of switches that route the traffic according to a fixed shortest path (XYDT based) discipline. It uses input buffering scheme and employs multi-class wormhole forwarding to support multiple service priority classes thus ensuring QoS.

I. INTRODUCTION

Modern system-on-chip (SOC) designs consist of numerous heterogeneous components integrated onto a single chip. As the number of components on a single chip and their performance continue to increase, the design of the communication architecture plays a major role in defining the area, performance, and energy consumption of the overall system.

To overcome these effects, SoCs commonly use Network-on-Chip (NoC), which offers attractive solutions to these problems. The NoC approach employs concepts from traditional communication networking to interconnect existing components reliably, in a plug-and-play fashion rather than point to point connectivity. The ability to reconfigure NoCs has attracted researchers to explore the mapping of Spiking Neural Networks (SNN) to NoC. SNNs are more powerful than conventional artificial NN models as information is transmitted by the means of pulses or spikes. However, as the architecture grows dense, there is a linear growth of the area occupied by the routers with the SNN interconnect requirements. It also poses new challenges in exploring solutions to traditional networking problems such as routing, quality-of-service, and flow and congestion control.

This has fostered the need to create multicast groups where many neurons are assigned a single unique router address within the architecture which will allow reduction of the overall SNN area requirements. Based on the above considerations, we propose an architecture named MAQNIS (MApping QoS into Noc Incorporated Snn) As a typical

guideline we classify system traffic into four common classes of service: Signaling, Real-Time, RD/WR and Block-Transfer. We model traffic behavior for each class and define the Quality of Service requirements of each class in terms of throughput, end-to-end delay and relative priority. Section 2 reviews previously proposed SNN based NoC architecture and the current challenges. Section 3 presents the overview of the MAQNIS architecture and Section 4 presents a detailed discussion of the proposed architecture. Finally the future works are discussed in Section 5.

II. BACKBONE DESIGN

A. Network On Chips

The NOC architecture is an $m \times n$ mesh of routers, and resources placed on the slots formed by the routers called the “core” as shown in figure 1. Each router is connected to one resource and four neighboring routers, and each resource is connected to one router in a regular 2-D mesh pattern. The communication links of the NoC are shared by many signals. Therefore, as the complexity of integrated systems keeps growing, an NoC provides enhanced performance (such as throughput) and scalability in comparison with traditional communication architectures.

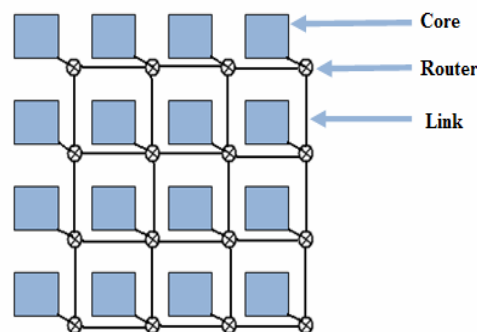


Figure 1: 2-D Regular mesh NoC

B. Bio-inspired Paradigm

Spiking Neural Networks (SNNs), which interacts using pulses or spikes, emulate more closely real biological neurons of the Brain. Consider the interconnectivity requirements of a feed-forward (FF), 2-layer $n \times m$ SNN network with each neuron in layer 1 connected to m neurons in layer 2; see Figure 2(a). [2]

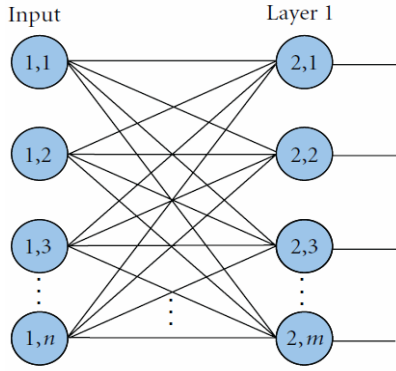


Figure 2(a): 2-layer SNN Network

C. Snn Based Noc Architecture

NoC strategy uses individual routers to group n synapses and the associated neuron, using a novel structure referred to as a neural tile, illustrated in Figure 3. The neural tile includes a NoC router and neuron/synapse elements. Communication between neural tiles is performed via router packets on North, South, East or West (N, S, E or W) neural tile ports. NoC spike packets contain destination neuron/synapse address. Data packets received by a neural tile, other than the intended destination tile, are passed directly through the NoC router. Spikes are generated and are routed to current tile. If the packet destination tile address matches the current neural tile address, a spike is generated to the addressed synapse within the current neural tile. [5]

D. Current Challenges

Although NoCs can borrow concepts and techniques from the well-established domain of computer networking, it is impractical to blindly reuse features of "classical" computer networks and symmetric multiprocessors.

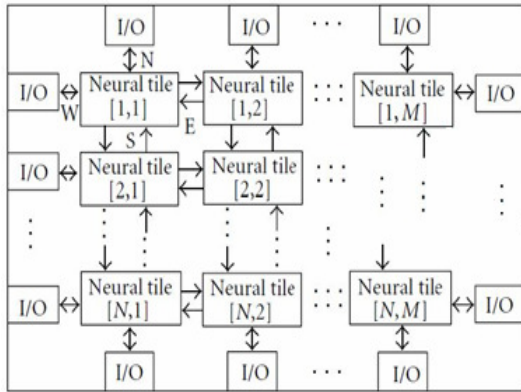


Figure 3: Existing Architecture overview

A NoC design is best used only in a high-bandwidth, large scale environment. In particular, NoC switches should be small, energy-efficient, and fast. Many researchers think that

NoCs need to support quality of service. This is an intentional flaw because it is believed with a good locality (heterogeneous NoC) and a good routing algorithm; a very good Best-Effort system can be build. [6]

Further implementing SNN into NoC has got a big problem of how to optimize the tile area and creation of dense architecture. For example, optimizing the numbers of neurons and synapses within individual neural tiles will reduce the number of routers required for the SNN implementations; this has the effect of creating multicast groups where many neurons are assigned a single unique router address within the architecture. This strategy will allow reduction of the overall SNN area requirements. There is also a linear growth of the area occupied by the routers with the SNN interconnect requirements.

III. MAQNIS: OVERVIEW

The MAQNIS architecture is illustrated in Figure 6 as a 2-dimensional array of interconnected neural tiles surrounded by I/O blocks. The neural tiles are connected in North, East, South, and West directions forming a nearest neighbor connect scheme. Each neural tile can be programmed to realize neuron-level functions which collectively implement an SNN. An SNN that is trained using backpropagation is realized on the MAQNIS architecture by programming the tile functionality and connectivity.

A. 2-D Irregular Mesh Noc

We choose a simple irregular mesh interconnection topology as basic topology, because it is simplest from a layout perspective and the local interconnections between resources and routers are independent of the size of the network. Moreover, routing in a heterogeneous network is easy resulting in potentially small switches, high capacity, short clock cycle, and overall scalability.

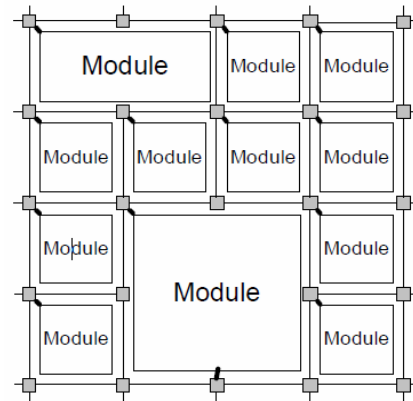


Figure 4: A simple 2-D irregular mesh NoC

The modified NOC consists (Figure 4) of resources and routers that are connected using channels as a mesh (Manhattan like structure) so that they are able to communicate with each other by sending messages. Each

router is connected to four other neighboring routers through input and output channels. A channel is a two one directional point-to-point link between two routers or a resource and a router. Routers may have internal queues to handle congestion. Our QoS network architecture is based on wormhole packet routing which reduces latency and buffer requirements in the routers.

B. Neuron Interconnectivity

The neural tile is viewed as a macro-block of MAQNIS. As our architecture is based on irregular mesh topology, each neural tile has different number of synapses compared to the other. The router of each tile is used to communicate spike events and the data to its group of n synapses, based on the quality of service levels. This enables a reduced number of connections; for example, an SNN interconnect density of 106 can be implemented using 4×103 . The output responses from each synapse cell are connected together to produce the desired response from the point neuron. The bandwidth of each inter-router link is similarly adjusted to accommodate the expected traffic and fulfill QoS requirements at the specific link.

B. Spike Event Communication

The inputs and outputs of the synapse cells are controlled via the router. The events of a spike train are received as data packets from neighboring routers where each spike-event packet includes a source address and destination address. Figure 5(a) illustrates the packet format where the header contains information on the type of mode of the data contained in the payload (i.e., runtime, configuration, or error) and the target address. The command field identifies the payload (quality of service levels), specifying the type of operation. The rest is an arbitrary length payload that includes operation-specific control information such as sender identification and the actual data to be transmitted.

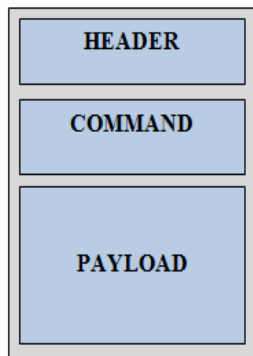


Figure 5 (a): Packet Format

The packet is divided into multiple flits. Flit transfer over the link is controlled by handshake. The flits are classified into the following types:

- FP (Full Packet): A one-flit packet
- EP (End of Packet): Last flit in a packet

- BDY (Body): A non-last flit

Thus, all but the last flit in a packet are tagged BDY. The first flit of a packet can be detected as the first valid flit following either a FP or EP flit (this identification triggers the routing mechanism). Each flit carries their own identity to which packet they belong and also contains information about the service levels as depicted in figure 5(b).

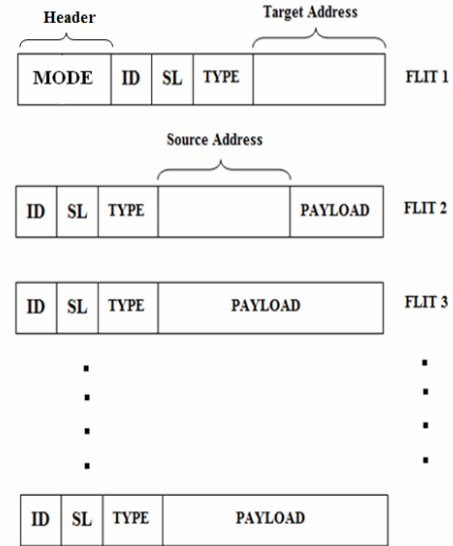


Figure 5 (b): Flit Format

Tile routers send an output packet each time a spike-event occurs within the tile. Exploiting the relatively low frequency of biological spike trains, MAQNIS can use a regular and scalable NoC structure. The time-multiplexing of spike data along router paths between SNN layers enables large parallel networks and high levels of routing ability, without the need for an overwhelmingly large number of connections.

C. Quality Service Levels

MAQNIS identifies four different types of communication requirements and define appropriate service levels (SL) to support them namely Signaling, Real-Time, Read/Write and Block-Transfer.

1. **Signaling** covers urgent messages and very short packets that are given the highest priority in the network to assure shortest latency.
2. **Real-Time** service level guarantees bandwidth and latency to real-time applications, such as streamed audio and video processing.
3. **Read/Write (RD/WR)** service level provides bus semantics and is hence designed to support short memory and register accesses.

4. **Block-Transfer** service level is used for transferring long messages and large blocks of data and is given the lowest priority.

A priority ranking is established among the four service levels on receiving the spike input and verifying the SL bit. Additional service levels may be defined if desired, as long as a priority ranking is adhered to.

IV. MAQNIS ARCHITECTURE

The MAQNIS neural tile is illustrated in Figure 6 and highlights the connections between the QoS based NoC router, synapse cells and neuron. The packet-switched router implements communication paths with buffer support.

A. Router Interconnectivity

Routers connect to up to five links, designed for planar interconnect to four mesh neighbors and to one chip module. Every arriving flit is first stored in an input buffer. On the first flit of a packet, the router determines to which output port that packet is destined. The router then schedules the transmission for each flit on the appropriate output port. The router uses an XY-DT routing scheme where among all Shortest Paths between each Source -Destination pair, it selects a path that makes a minimal possible number of routing steps which deviate from XY routing policy[4]. This leads to a simple, cost-effective and efficient router implementation in case of irregular mesh topology and it involves the following buses/lines

1. *SPIKE I/P* initiates a spike on an individual synapse cell.
2. *SPIKE O/P* receives spike events from the neuron.
3. *MODE* specifies the tile operation, namely, runtime or configuration programming.
4. *Indexing* bus is used to address individual synapse cells (via address decoders) for receiving spike events or configuration data.
5. *ACK* acknowledges the correct synapse addressing.
6. The *Config Data* bus is used to transmit configuration data to the cells.
7. *SL* is used to specify the flit service level.
8. *Type* specifies the type of flit (FP, EP and BDY).
9. *Buffer Credit* indicates the availability of buffer space for one flit at each specified service levels at the input port of the next router.

Apart from these communication lines, the router makes use of three tables

Address table (AT): Routing information for each flit (flit id, source and target address) is stored in the AT until the tail flit of the packet is received, processed and delivered.

Buffer State Table (BST): Each output port maintains the number of available flit slots per each service level in the buffer of the next input port. These numbers are stored in this table.

Serviced Number Table (SNT): The present state of round-robin scheduling is stored in SNT table for each service level at each output port.

B. Maqn timer Communication

1) Reconfigurable And Runtime Operations

The neural tiles operate in one of two modes: runtime or configuration. This is defined by the information in the packet header where 0 and 1 define the payload to be either runtime or configuration, respectively. In runtime mode, MAQNIS routes spike events (simply data packets) and the payload does not exceed more than two flits. In configuration or programming mode, MAQNIS is configured to realize particular synapse models and the desired SNN topology. Configuration data is delivered in the form of data packets where each packet is addressed to a particular neural tile and contains information on the configuration of the router's AT, the selection of cell synapse weights via the programmable voltage lines, V_q , and other neural tile parameters. In this mode, a packet typically contains multiple.

2) Neural Tile

There are separate buffers for each of the four service levels ("direct buffer mapping") at the input and the output ports of the router as shown in figure 8. Relatively small buffers are allocated to each service level, capable of storing only a few flits (this is a tunable design parameter).

The two major functionalities of a router are

- (i) Transmission of data packets
- (ii) Selection and configuration of synapse cells

(i)Transmission of data packets

Each neural tile has a unique address, and the synaptic connectivity is specified using an Address Table (AT) within each router. Tile routers send an output packet each time a spike-event occurs within the tile. The events of a spike train are received as data packets from neighbouring routers where each spike-event packet includes a source address (indicating the SNN neuron/tile in which the event originated) and destination address (of neural tile and synapse). Based on the SL of the received data packet, it is stored in the respective buffer at the input port of the router. On receiving the first flit, the router sends an ACK signal and stores the routing information. When a flit is forwarded from an input to an output port, one buffer becomes available and a buffer-credit is sent back to the previous router on separate out-of-band wires.

The output port maintains the number of available flit slots per each service level in the buffer of the next input port. These numbers are stored in the Buffer State Table (BST). The number is decremented upon transmitting a flit and incremented upon receiving a 'buffer credit' from the next router.

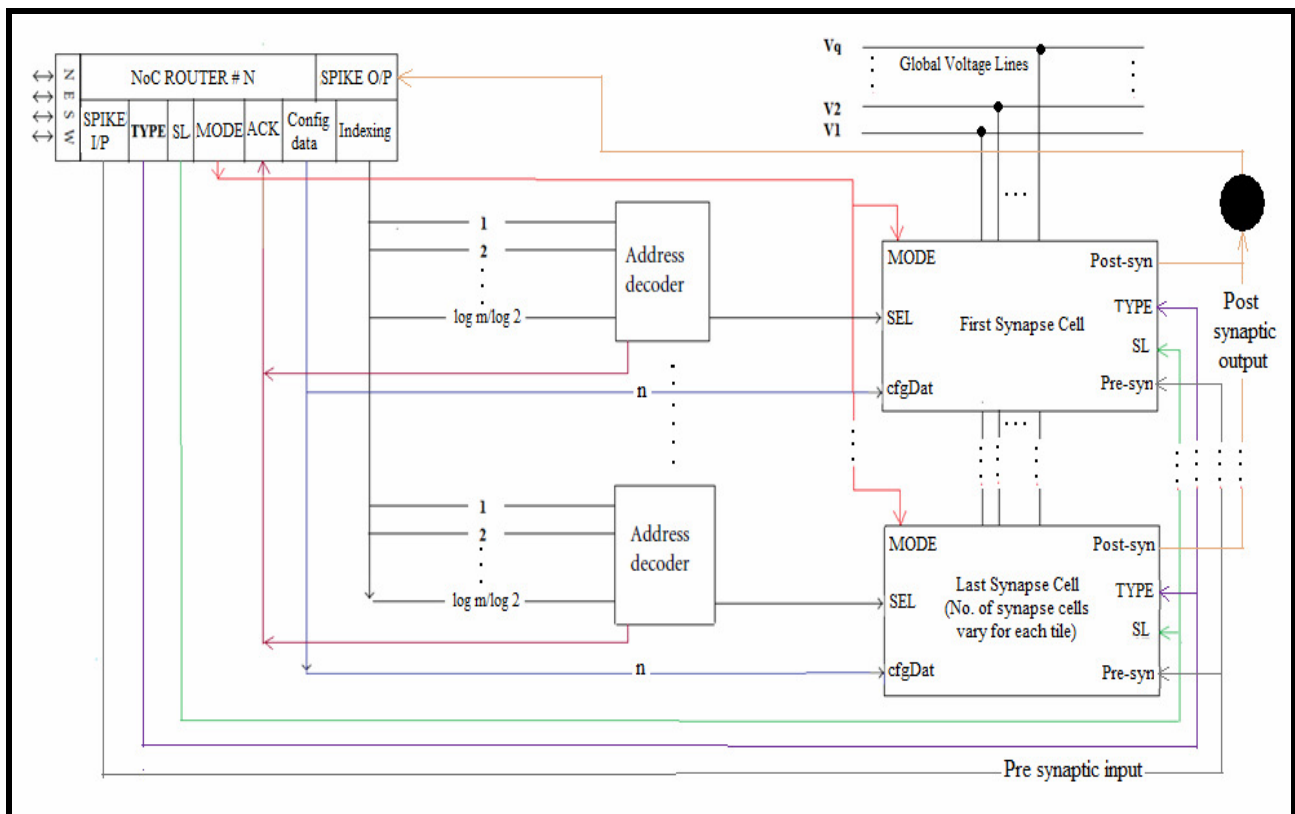


Figure 6: MAQNIS architecture

Here a preemptive priority based scheduling technique is employed for the transmission of flits of different service levels. However a round robin scheduling is used within the same service levels. The present state of round-robin scheduling is stored in the State Number Table (SNT) table for each service level at each output port.

(ii) *Selection and configuration of synapse cells*

When a flit is received, the router sends it along with the spike to any one of the multiple synapse cells. Each synapse cell is hardwired to q programmable weight voltage lines during configuration. The configuration data of the flit is responsible for deciding the spike output based on the net voltage. The flit with the highest spike output will be directed from that synapse cell to the point neuron along with its spike as SPIKE O/P of the neural tile. The basis for SNN is the strengthening and weakening of synaptic weights, whereby training algorithms could be used to derive appropriate weight values to reflect a mapping between the input and desired output data. Faults occurring in individual synapses could be tolerated by using such algorithms to appropriately retrain the network when the output deviates from the “marked” patterns. Now when a spike event is detected at the Spike O/P, the AT identifies which target neural tiles and synapses must receive event notification.

However starvation may occur due to prolonged buffering of the data packets having low priority SL. Hence, a timer is used which continuously monitor the waiting time of the flits in the buffer. When the timer indicates the prolonged buffering, the transmission of current packet is preempted and the starved packet gets through. The configuration data is set accordingly to increase the spike output. After servicing this packet, the processing of the preempted packet is resumed.

V. CONCLUSION AND FUTURE WORK

A novel MAQNIS architecture has been presented which utilizes quality of service based NoC routers and novel synapse cells to provide the programmable and scalable interconnect and bio-computational resources, required in large scale SNNs. Overall, the approach has been proven to be very promising for reducing the area requirements and reducing spike-event traffic.

The concept of preemption causes stalls not only in the router where different service-level flits contend, but also in other routers on the preempted packet route. In the latter routers, the output ports are busy and idle, waiting for data, even though there might be flits of the same service level from other input ports that are ready for sending. In future, employing virtual channels for each service level will allow

better utilization of the output ports and links. Future work will also explore efficient training algorithm for SNN functionality.

REFERENCES

- [1] W.J. Dally, B. Towles, Route packets, not wires: on-chip interconnection networks, DAC 2001, June 18–22, 2001, Las Vegas, Nevada, USA.
- [2] W. Maass, "Computation with Spiking Neurons: The Handbook of Brain Theory and NNs", 2nd Ed, MIT Press, 2001
- [3] A Network on Chip Architecture and Design Methodology, Proceedings of the IEEE Computer Society Annual Symposium on VLSI, p.117, April 25-26, 2002
- [4] E. Bolotin, et al., "Efficient Routing in Irregular Topology NoCs", CCIT Report #554, EE Dept, Technion, Sep. 2005
- [5] Jim Harkin, Fearghal Morgan, Liam McDaid, Steve Hall, Brian McGinley, Seamus Cawley, "A reconfigurable and biologically inspired paradigm for computation using network-on-chip and spiking neural networks", International Journal Of Reconfigurable Computing, Volume 2009, January 2009.
- [6] Shashi Kumar, Axel Jantsch, Juha-Pekka Soininen, Martti Forsell, Mikael Millberg, Johnny Oberg, Kari Tiensyrja and Ahmed Hemani, "A Network on Chip Architecture and Design Methodology", Proceedings of the IEEE Computer Society Annual Symposium on VLSI 2002 (ISVLSI02).
- [7] S. Furber, "Future trends in SoC interconnect," in *Proceedings of the IEEE VLSI-TSA International Symposium on VLSI Design, Automation and Test (VLSI-TSADAT '05)*, pp. 295–298, Hsinchu, Taiwan, April 2005.
- [8] T Felicijan, S.B. Furber, "An Asynchronous On-Chip Network Router with Quality-of-Service (QoS) Support," Proc. Of IEEE Int. SOC Conf., Santa Clara, CA, pp. 274-277, 2004.
- [9] R. Dobkin, V. Vishnyakov, E. Friedman and R. Ginosar, "An Asynchronous Router for Multiple Service Levels Networks on Chip," Proc. ASYNC, pp. 44-53, 2005.
- [10] Yiping Dong, Ce Li, Zhen Lin, and Takahiro Watanabe, "Multiple Network-on-Chip Model for High Performance Neural Network," Journal Of Semiconductor Technology And Science, Vol.10, No.1, March, 2010.