A High Performance Digital Neural Processor Design by Network on Chip Architecture

Yiping, Dong Ce, Li Hui, Liu Takahiro, Watanabe
Waseda University, Japan

ABSTRACT

This paper describes a high performance neural processor by using a Network on Chip (NoC) architecture to solve the interconnection and performance problems in hardware neural networks. The proposed NoC-based neural processor is composed of 20 tiles in 4x5 2-D array, and each tile includes a Process Element (PE) and a packet switched router. In each PE, four neurons are implemented to achieve low communication load. The network is 2-D torus topology, and it has a 32 G/s bandwidth and asynchronous clocking system. Our proposed neural processor is designed using 90-nm CMOS technology with one Poly and nine metals, and its performance is evaluated. As a result, it can achieve over 3.1 G Connection Per Second (CPS) of performance while power dissipation is 1.1317 W at 1.2 V supply-voltage and 25 mm^2 chip area. Compared with the other existing hardware neural networks, the proposed processor can achieve low communication load and high performance, and it is reconfigurable and extendable.

1. INTRODUCTION

Digital architectures of Artificial Neural Network (ANN) have critical implementation issues such as cost, interconnection and performance. In general, architecture of a digital ANN is application specific, so that it is inevitably re-designed for different applications and a network topology suitable for the application should be adopted to avoid the complex interconnection between layers. The heavy communication load is also a more serious problem when the number of neurons becomes larger.

Recently, Network on Chip (NoC) using an on-chip packet-switched interconnection network replaces global shard buses with point-to-point interconnection network as a smart data transmission system. The packet-based network with high-level parallel architecture has attracted much attention of a lot of research groups to solve complex on-chip interconnection problems of a large system-on-chip (SoC) [1].

In this paper, a novel neural processor design based on NoC architecture is described. We call this new type processor "NoCNN". The basic module of NoCNN is a tile, which is composed of one PE (Processing Element) and one 1-GHz 5-port wormhole-switched router, and those tiles are arranged in 2-D array. In each PE, four neurons are integrated. We integrate twenty such tiles as the whole NoCNN because twenty tiles are enough to implement various applications and a torus topology is adopted for its simplicity. Simulation results show that our proposed NoCNN can overcome the problems of an existing digital ANN about reconfigurability, design cost, interconnection and performance [2].

The remainder of this paper is organized as follows. Section 2 gives an overview of the related work and our design motivation. Section 3 describes the NoC architecture to structure NoCNN. Section 4 presents performance evaluation and implementation results. Section 5 concludes by summarizing NoCNN.

2. RELATED WORKS

The digital architecture of ANN has a high precision and good extensibility, and a lot of EDA tools are provided to support the digital implementation [3]. Different kinds of digital architectures were proposed for ANN, such as systolic array devices, slice architecture, single instruction multiple data (SIMD), and so on [2]. These architectures make possible quick development of hardware ANN. However, there exist some drawbacks. The reconfigurability of systolic array architecture and performance of the slice architecture are not well. Although the SIMD architecture has a little improvement in them, the complex mapping method limits it [2]. Furthermore, all of these architectures suffer from the interconnection problem because of the global shard buses and point-to-point interconnection.

These years, a lot of research groups contributed to improve the digital ANN. However, the design method is still based on the existing architectures. A new design method with new architecture is required to overcome the problems of interconnection, performance, reconfigurability and so on. NoC is such an architecture that can solve the communication problem and support high performance for large size SoC design. The Intel 80-tile NoC chip [4] is one of good design example for high performance SoC. Therefore, this paper focuses on proposing a new design method with NoC architecture to build a new type of digital ANN to overcome the drawbacks in the traditional ANNs and achieve the higher performance.

3. NOC ARCHITECTURE FOR NEURAL PROCESSOR

The proposed NoCNN contains 20 tiles arranged in a 4x5 2-D torus network. In this section, the key components of this NoC and the packet format are described in detail. One neuron is designed and then four neurons are integrated in one PE (Processing Element), and the router is designed to connect with PE as one tile. Finally, such 20 tiles are arranged in 2-D array and connected with each other in 2-D torus topology to build the whole NoCNN.

3.1 PE Architecture

The most common ANN is a multi-layer perceptron. We know that one hardware neuron requires a high performance and high precision multiplication block for computation and memory block for holding weight value. Besides, an adder block and activation function block are also required [5]. RAM is used to store weight values of this neuron and controlled via the weight address. The input data are selected by MUX (multiplexer) and then multiplied with weight value by the multiplier. When all the products have been added, this sum will be used for searching output via activation function. The activation function is implemented by means of a lookup table (LUT) which is stored in RAM. 32-bit fixed-point architecture is used for our design to suit for the requirement of high precision. One signal bit, six integer bits, and twenty-five fraction bits can cover the range of [-64,64) with a quantization error of 2.98023224E-8.

978-1-4244-8499-7/11/$26.00 ©2011 IEEE

In our architecture, four neurons in the same layer in an ANN are integrated in one Processing Element (PE) to reduce the total transmission packets and communication load. These four neurons share one LUT to reduce design cost. The PE also requires a decoder, an encoder, a controller and a weight address generator. When an input packet arrives at a PE, a decoder decodes the address of the neuron to be used and transmit the decoded input to each neuron in this PE. When a controller receives the decoded address information from a decoder, it controls the weight address generator to generate the virtual address and transmit to each neuron. Then the generated address will control the RAM which stores the weight values. When all neurons in this PE complete their calculation and LUT task, their outputs are transmitted to an encoder. It holds outputs as one single packet. In this packet, each payload part comes from each neuron and header part comes from one RAM.

More than four neurons can be designed in one PE to reduce the communication load more, whereas the system performance may decrease. Also we observe that that total number of neurons of a general digital ANN [5] is always every fourth such as 4, 8, 12, and so on. Therefore, four neurons in one PE could achieve the better balance between performance and communication load. We will explain the reduction of communication load in Sect. 4.

3.2 Router Architecture and Packet Format

A 1GHz 5-port wormhole packet-switched router of 32 GB/s bandwidth is designed for the proposed 20-tile NoCNN. Each port has two links for dead-lock free routing and reducing the latency. A block diagram of this 4-stage pipeline router and pipeline stages are shown in Fig. 1. Five input ports are connected with four routers and one PE. Each flit needs to be processed through the steps of Routing Computation (RC), Virtual-channel Allocation (VA), Switch Allocation (SA) and Switch Traversal (ST). Assume that a header flit arrives at an input port, and it needs to be decoded and buffered according to the flit information by RC block in the first stage. The header flit then chooses a channel by VA block in the second stage. The flit travels through the selected virtual-channel and chooses an proper output port by SA block in the third stage. In this stage, each SA block will get 5-bit input from each input port and send 5-bit output signal to switch crossbar to control the output port. Each SA block controls one output port. Flit travels through the crossbar switch in the fourth stage.

The switch allocator consists of three parts: decoder, arbiter and hold logic. In the decoder part, 5-bit from header of input packet is decoded. 2-bit is used for partition the type of the flit, 3-bit for output choice. The second part is a fixed-priority arbiter. The port-4 which connects with PE will get a high priority, because the time of data in PE is longer than in router transmission. The hold logic part is designed to hold the selected port for the payload.

Each packet of NoC contains one header and some payloads, and the number of payload is dependent of the number of neurons that used in this PE which connects with this router. Thus the number of payload is always 1-4. The header information contains 2 bits for VCN (Virtual Channel Number), 2 bits for FT (Flit Type), 4 bits for UN (Used Neuron), 15 bits for PCI (PE Control Information), and 3 bits for each DA (Destination Address). One packet can store four DAs. The payload contains 2 bits for FT and 32 bits for DFN (Data from the Former Neuron).

3.3 Architecture of 4x5 NoCNN processor

The proposed 4x5 NoCNN is constructed by a building block style. The key component is a set of PE and a router, which are arranged

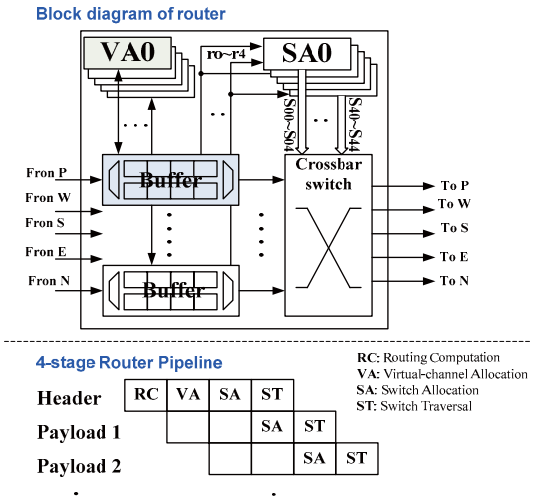

FIGURE 1. BLOCK DIAGRAM OF ROUTER AND PIPELINE PROCESSING.

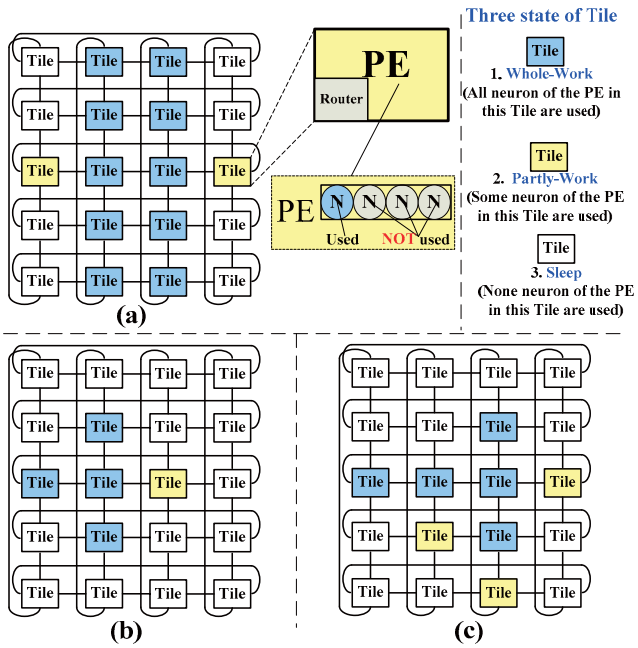

FIGURE 2. 4X5 NOCNN PROCESSOR FOR THREE ANN APPLICATIONS. A)3-20-20-1 FF-ANN B)4-12-1 FF-ANN C)4-7-13-1 FF-ANN.

in 4x5 2D torus topology. Some other networks (e.g., flat butterfly or Clos) might be substantially lower latency compared with proposed torus topology, but they will limit the reconfigurability and extensibility of the whole system. The placement of each neuron is fixed and the transmission paths are also fixed. Figure 2 shows some implementations for three real applications of ANN [6-8].

In Fig. 2, three colors indicate different states of tiles, respectively. Blue Whole-Work tile means that all neurons in the PE of this tile work, yellow Partly-Work tile means that some of neurons in the PE of this tile work, and white Sleep tile means any neuron in the PE of this tile does not work. For example, Fig. 2(a) is an implementation of ANN with 3, 20, 20 and 1 neuron in each layer. One PE in the first column of the network is used as input layer (three neurons in this PE work, the rest one neuron does not work), five PEs of blue tiles in the second column of the network are used as first hidden layer, five PEs in third column of the network are used as second hidden layer, one PE in fourth column of the network is used as output layer (one

neuron in this PE work, other three neurons need not work). Figure. 2(b) and (c) show that the other two applications using the same implementation method. The 4x5 NoCNN can implement different applications with at most 20-20-20-20 architecture. If more neurons are required for another application, the network topology should be extended by adding the tiles and the packet information of address control may be a little different. But, the implementation method is almost same.

4. Implementation Results and Performance Evaluation

The 4x5 NoCNN processor is designed using 90-nm CMOS process technology with one Poly and nine metals. The functional blocks of the chip and individual tiles are layouted as shown in Fig. 3. The 25 mm2 fully custom design contains 61.33 million NAND2 based cells. Each 1.152 mm2 tile contains 3.023 million cells. It dissipating 1.1317 W of power at 1.2 V supply.

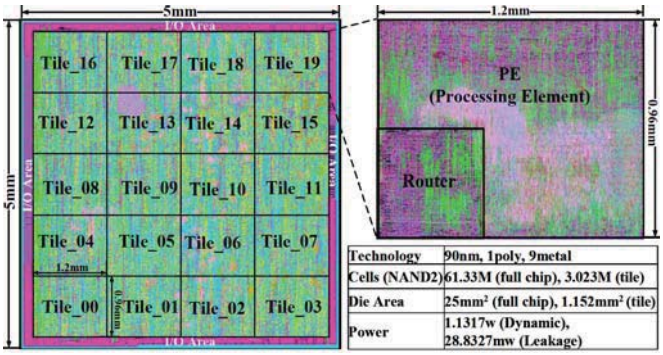

Figure 3. NoCNN processor layout design.

The features of the proposed NoCNN processor are analyzed and the performance parameters are evaluated, as follows.

4.1 Reconfigurability and Extensibility

NoCNN based on the proposed NoC architecture is reconfigurable. The weight values and activation functions cans be easily changed for different applications because they are stored in RAMs. Also, different topology of ANNs can be implemented on the same processor which is controlled by sending new packets. NoCNN is also extensible. We use a hierarchical design flow. The key components of PE and router are designed separately to compose a tile of this processor. In this work, NoCNN is constructed by such 20 tiles and arranged in 4x5 torus topology. It could easily add or delete any tile to meet the requirement of ANN.

These reconfigurability and extensibility are remarkable features to make NoCNN implementation possible for different applications without any hardware change, while the other existing hardware ANNs must be changed for each application. Fig .2 shows the optimization mapping and configuration of three ANN applications.

4.2 Communication Load Evaluation

Traditional digital ANNs use bus-based point to point (P2P) architecture for data transmission. However, the proposed NoCNN uses NoC architecture to transmit the data by routers. We assume each neuron in the same layer needs to transmit 1 packet/pattern to each neuron of the next layer. The communication load of three real applications (mentioned in Sect. 3.3) are implemented by using the proposed NoC structure and compared with a general P2P bus architecture, as shown in Table 1. The first application of ANN has 3, 20, 20 and 1 neurons in each layer. With the P2P architecture, the number of packet is 480 (3x20+20x20+20x1) and each packet has the same bit size of 32. Thus the total communication load is 15360 bits/pattern. If we use the NoC architecture, the packet number and packet size are decided by the number of tiles and the number of neurons in each tile. From Fig. 2(a), the tile for input layer need transmit 5 packets to the next hidden layer and the packet size is 136 bits/pattern (34 bits x (1 header + 3 payloads)). The other layers follow the same rule. The total communication load is 5780 bits/pattern (136 bits x 5 + 170 bits x 30).

Table 1 Comparison of communication load

Topology	Type	Packet number	Packet size (bit)	Total size (bit/pattern)
3-20-20-1	P2P	480	32	15360
	NoC	35	136,170	**5780**
4-12-1	P2P	60	32	1920
	NoC	6	170	**1020**
4-7-13-1	P2P	132	32	4224
	NoC	14	68,136,170	**2142**

As shown in Table 1, the communication load of total packet size per pattern can reduce by 46.9%-62.4% using the proposed NoC architecture, compared with P2P architecture. This reduction owes to packing neurons and packet-based data transmission mechanism.

4.3 Performance Evaluation

Three applications described in Sect.3.3 are simulated on 4x5 NoCNN by using NIRGAM NoC simulator [9]. Latency of each application is shown in Fig. 4.

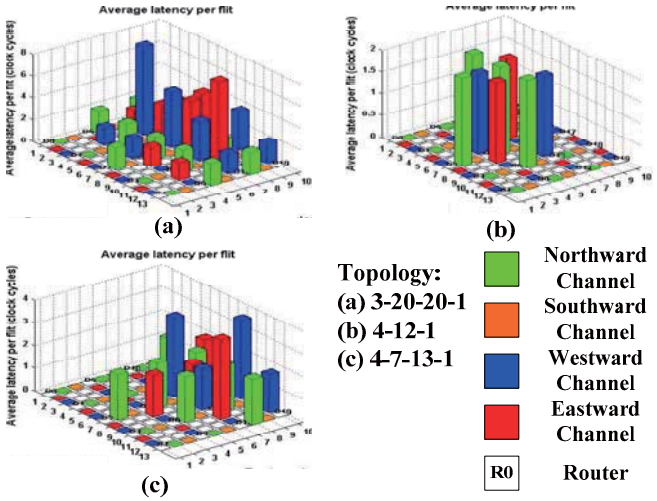

Figure 4. Average latency per flit of three applications.

The critical path of the first application is appeared in data transmission from the first hidden layer to the second hidden layer. It costs 37.06 cycles. And the frequency of the proposed system is 1GHz, thus the system running time of the first application is 37.06ns. The system running time of the second and the third applications

Table 2 Comparison of CPS and CPSPW.

Name	Architecture	Precision	Neurons	CPS				CPSPW		
				Max	App1*	App2	App3	App1	App2	App3
Lneuro-2.3	Slice	16, 32b	12PE	720M	440M	340M	300M	0.92M	5.76M	2.27M
NC3001	SIMD	32b	1,32	1G	344M	177M	195.31M	0.72M	2.95M	1.48M
NM6403	SIMD	64x64b	1, 64	1.2G	206M	106M	117.2M	0.43M	1.77M	0.89M
MA-16	Systolic array	16b	16PE	400M	na**	na	na	na	na	na
4x5 NoCNN	NoC	1-32b	20PE	3.1G	1.62G	557M	867M	3.37M	9.29M	6.57M

* Application 1; ** not available;

are 7.664ns and 22.163ns, respectively. From the results, we find that the latency is decided by topology, especially the number of the neurons of an input and hidden layer.

The most common performance measurement is the Connection-Per-Second (CPS), which is defined as the rate of multiplication and accumulation operations. However, some systems could get a high performance according to the large number of neurons. So that, the value of CPS is normalized by dividing it by the number of weights, that is, the connection per second per weight number (CPSPW), the value of CPS and CPSPW for three applications are compared between our proposed NoCNN and other existing digital ANNs [5][10] as shown in Table 2.

Table 2 shows a Comparison of CPS and CPSPW for various ANNs, and note that PE of different system has different architecture. MA-16 is a digital ANN with systolic array architecture, and it could not implement three applications due to the limit number of neurons. Compared with other digital ANNs, the proposed NoCNN has the highest performance of CPS and CPSPW for different applications. It increases CPS and CPSPW at least 63.8% and 61.3%, respectively. We analyze these results as follows: compared with the slice architecture (e.g. Lneuro-2.3) which used a bus control, the NoC control mechanism is more suitable for the Multi-Processor SoC; the digital ANN with SIMD architecture (e.g. NC3001 and NM6403) can just implement ANN layer by layer, whereas out NoCNN could implement different layers' neurons at the same time.

Since higher CPS may also result high speed, the system running time of NoCNN for three different applications is evaluated. Figure 5 shows the result, where system running time means average time per pattern to complete all transmission. From Fig. 5, NoCNN obviously shows the lowest system running time compared with other four digital ANNs.

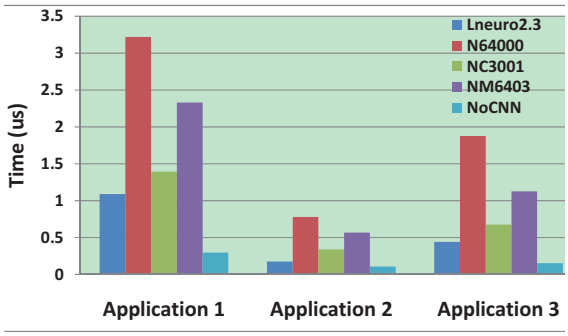

Figure 5. Comparison of system running time.

5. Conclusion

This paper presents an NoC architecture to build a 20-tile reconfigurable and high performance NoCNN processor, and evaluates the processor by implementing with a 90-nm process technology. It could achieve 3.1G CPS, while power dissipation is 1.1317 W of power at 1.2 V supply voltage and 25 mm^2 chip area. The simulation results indicate that our proposed NoCNN is effective in increasing performance and reducing communication load. These results also demonstrate the feasibility for high performance and low communication load by NoC-based digital ANN.

Acknowledgments

This research was partly supported by Waseda University Global COE Program `International Research and Education Center for Ambient SoC' sponsored by MEXT, Japan (GCOE), and also partially supported by Regional Innovation Cluster Program 2nd Stage by MEXT, and Core Research for Evolutional Science and Technology (CREST), Japan Science and Technology.

References

[1] L. Benini and G. D. Micheli, *Networks On Chips: Technology and tools*. Morgan Kaufmann, 2005.

[2] C. S. Lindsey, "Neural networks in hardware: Architectures, products and applications," in lecture notes of Neural Networks, Aug. 2002.

[3] L. Gatet, H. Tap-Beteille, and F. Bony, "Comparison between analog and digital neural network implementations for range-finding applications," IEEE Trans. Neural Network, vol. 20, no. 2, pp. 460–470, Mar. 2009.

[4] S. Vangal and et al., "An 80-tile sub-100-w teraflops processor in 65-nm cmos," IEEE Journal of Solid-State Circuits, vol. 43, no. 1, pp. 29–41, 2008.

[5] B. Muller, J. Reinhardt, and M. Strickland, Neural Networks: An Introduction (Physics of Neural Networks). Springer, 2002.

[6] A. Ouchar, R. Aksas, and H. Baudrand, "Artificial neural network for computing the resonant frequency of circular patch antennas," Microwave and optical technology letters, vol. 47, no. 6, pp. 564–566, Oct. 2005.

[7] H. M. Yao, H. B. Vuthaluru, M. O. Tade, and D. Djukanovic, "Artificial neural network-based prediction of hydrogen content of coal in power station boilers," Fuel, vol. 84, no. 12–13, pp. 1535–1542, Sep. 2005.

[8] M. Rajendra, P. Jena, and H. Raheman, "Prediction of optimized pretreatment process parameters for biodiesel production using ann and ga," Fuel, vol. 88, no. 5, pp. 868–875, 2009.

[9] L. Jain, "Nirgam," in University of Southampton UK, http://www.nirgam.ecs.soton.ac.uk.

[10] F. M. Dias, A. Antunes, and A. M. Mota, "Artificial neural networks: a review of commercial hardware," Engineering Applications of Artificial Intelligence, vol. 17, no. 8, pp. 945–952, Aug. 2004.