BiNMAC: Binarized neural Network Manycore ACcelerator

Ali Jafari
Dept. of Computer Science and Electrical Engineering, University of Maryland, Baltimore County
Baltimore, Maryland

Morteza Hosseini
Dept. of Computer Science and Electrical Engineering, University of Maryland, Baltimore County
Baltimore, Maryland

Adwaya Kulkarni
Dept. of Computer Science and Electrical Engineering, University of Maryland, Baltimore County
Baltimore, Maryland

Chintan Patel
Dept. of Computer Science and Electrical Engineering, University of Maryland, Baltimore County
Baltimore, Maryland

Tinoosh Mohsenin
Dept. of Computer Science and Electrical Engineering, University of Maryland, Baltimore County
Baltimore, Maryland

ABSTRACT

This paper presents a low power, domain-specific manycore accelerator referred to as "BiNMAC"- Binarized neural Network Manycore ACcelerator, which effectively maps and executes Binary Deep Neural Networks (BNNs). With only 2.40% and 1.88% area and power overhead, novel instructions such as *Population-Count* and *Patch-Select* are added to the ISA of the BiNMAC, each of which replaces frequently used functions that would have taken 52 and 4 clock cycles respectively with 1 clock cycle. A 64-cluster architecture of the BiNMAC is fully placed and routed in 65 nm TSMC CMOS technology, where a single cluster occupies an area of 0.53 mm^2 with a power of 223 mW at 1 GHz clock frequency. The 64-cluster architecture takes 36.5 mm^2 area and, if fully utilized, consumes a power of 16.4 W. We also propose a multilayer perceptron (MLP) neural network for multimodal time-series data classification. Binarized versions of the 3-layers MLP and ResNet-20 are implemented on BiNMAC. The implementation results show that BiNMAC consumes 0.02 mJ and 3.8 mJ energy which is 13× and 30× lower than implementation of standard non-binarized MLP and ResNet-20 on an equivalent predecessor platform. To compare the performance of the BiNMAC with other off-the-shelf platforms, the two networks are also implemented on the NVIDIA Jetson TX2 SoC (CPU+GPU). BiNMAC achieves 22× and 78× higher throughput and 23× and 41× lower energy consumption compared to TX2 SoC for the binarized MLP and ResNet-20, respectively.

ACM Reference Format:
Ali Jafari, Morteza Hosseini, Adwaya Kulkarni, Chintan Patel, and Tinoosh Mohsenin. 2018. BiNMAC: Binarized neural Network Manycore ACcelerator. In *GLSVLSI '18: 2018 Great Lakes Symposium on VLSI, May 23–25, 2018, Chicago, IL, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3194554.3194634

GLSVLSI '18, May 23–25, 2018, Chicago, IL, USA

ACM ISBN 978-1-4503-5724-1/18/05…$15.00
https://doi.org/10.1145/3194554.3194634

1 INTRODUCTION

In recent years, Deep Neural Networks (DNNs) have become increasingly popular because of their usability and outstanding results in areas such as computer vision, voice recognition, natural language processing, robotics and time-series data classification. Several methods have been proposed to address efficient training and inference in deep neural networks: Shallow networks, quantizing parameters, network binarization and compressing pre-trained deep networks. Although these methods improve the efficiency of DNNs for implementing on low-power embedded platforms, an off-chip memory is usually required to save large models. Therefore, power consumption is still high due to the need of constantly accessing the off-chip memory. In 2016, [1] proposed Binarized Neural Networks (BNNs) to address the previously mentioned challenges. BNN has a compact representation of network weights and activation values compared to a standard DNN by constraining each weight value to either -1 or +1. During the forward pass BNNs: 1) Drastically reduce the memory requirements. 2) Can eliminate the need of using off-chip memory. 3) Replace multiply and add operations with bit-wise XNOR and population-count operations. Since 2016, different works [1, 8] have shown that BNNs can achieve comparable accuracies to full-precision DNNs for some popular datasets such as MNIST, CIFAR10 and ImageNet. [10, 11] have proposed BNN hardware accelerators on CPU, GPU, FPGA and ASIC. Also, several research work has been carried out in domain specific platforms to implement simple processing cores for kernel-optimization rather than application-centric processors.

In this paper, we propose "BiNMAC"- Binarized neural Network Manycore ACcelerator, a novel energy-efficient programmable cluster-based manycore that is designed specifically for BNNs, implemented on 65 nm technology on chip area of size 36.5 mm^2, with a maximum total power of 16.4 W. The proposed architecture has novel instructions for different layers of BNNs, that reduce computations and consequently execution time. Additionally, a fully-connected BNN topology, is proposed which is designed for multimodal time-series data classification [3, 4]. We evaluate the functionality and performance of BiNMAC using binarazied versions of the fully-connected BNN and fully-convolutional Resnet-20.

2 BINARY DEEP NEURAL NETWORKS

Similar to standard neural networks, binary neural networks consist of convolution and fully-connected layers and activation function.

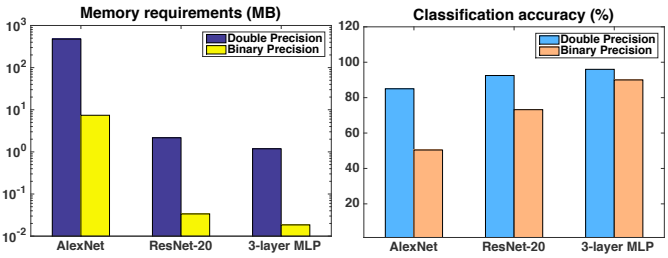

Figure 1: (A) Required memory (B) Classification accuracy for three NNs with double and binary precision weights.

However, these layers are implemented differently in BNNs: For the first layer of BNN, data from each input channel (e.g red, green and blue channels for colored images) are in full-precision values, but model weights of the BNN are constrained to either +1 or -1. Therefore, multiply-accumulate operations are replaced by a series of additions and subtractions. For the next layers, all input features are binarized using binary activation functions, and along with the binarized weights, the multiply-accumulate operations can be performed using only XNOR operations and population-count operation (that will be discussed in Section 3). In order to apply the activation function to the real-valued variables and transform them into either +1 or -1, deterministic or stochastic binarization functions are proposed by [1]. We use deterministic binarization, which is in fact a sign(x) function, since it is more efficient to implement on hardware. In this section, BiNMAC is evaluated using two different case studies including Physical Activity Monitoring and Image Recognition.

2.1 Case study 1: Physical Activity Monitoring

A multi-layer perceptron (MLP) neural network is proposed to classify multimodal time series signals. The proposed network has 3 fully-connected layers. Each layer in the network has a size of 256 nodes, and Sigmoid function is used as the activation function. We used TensorFlow framework to design and evaluate the MLP with a real-world dataset of Physical Activity Monitoring (PAMAP2) [9]. PAMAP2 records 12 physical activities performed by 9 subjects. The physical activities are, for instance: standing, walking, lying and sitting. Three inertial measurement units (IMU) and one heart rate monitor were used to record the data. In total, the dataset includes 40 channels of valid data. The network parameter size, with double precision values, is 1.2 MB. Binarizing the proposed MLP causes the parameter size to reduce by 64x. Also, the average classification accuracy over 8 subjects is 98%.

2.2 Case study 2: Image Recognition

Binarized ResNet-20 for CIFAR-10 image recognition dataset [6] is used to evaluate BiNMAC, without any change in the topology using the Torch implementation provided by [8]. Binarizing ResNet-20 network causes the parameter size to drop by 64x at the expense of 18.1% loss in classification accuracy. Fig. 1-A shows the required memory for three different neural network architectures, AlexNet [5], ResNet-20 [2], and the proposed MLP with binary and double precision weights and Fig. 1-B reflects their classification accuracy.

3 BINMAC MANYCORE ARCHITECTURE

3.1 BiNMAC Overview and Key Features

Fig. 2-A shows the block diagram of a 64 cluster version of the BiNMAC architecture, highlighting the processing cores in a single bus-based cluster. Fig. 2-B shows the post-layout view of a bus-based single cluster. Each processing core, cluster bus, cluster memory and the router was synthesized and fully placed and routed in a 65 nm CMOS technology using Cadence SoC Encounter and the post-layout implementation results for one cluster are summarized in Fig. 2-E. Each cluster comprises of 3 processing cores with 6 pipeline stages, a cluster memory of 3072 words (6KB), and low-latency bus interconnect for communication between cores within the cluster and a hierarchical routing architecture for intra-cluster communication. The processing core operates on a 16-bit data-path.

3.2 BiNMAC Manycore Platform Evaluation

To evaluate the functionality of BiNMAC, we developed a stand-alone simulator and a compiler in Java. This simulator provides cycle accurate results along with task execution and completion time, number of instructions, and memory usage per core. Each task of the neural network application, is first implemented in assembly language on each processing core in a cluster using the simulator. This code is given as an input to the simulator, which reads in the code. It then models the functionality of the processor and calculates the final state of register files and data memories. For execution time and energy consumption analysis of the algorithm, binaries obtained from manycore compiler are mapped on to hardware model of the manycore platform and simulated using Cadence NC-Verilog. The activity factor is then derived and is used by the Cadence SoC Encounter tool for accurate power computation.

3.3 BNN On-chip Memory Processing

In order to avoid use of an external DRAM, binary networks of proper size, whose network parameters and temporary feature map can fit inside the on-chip memory, are chosen. By doing so, not only the hardware overhead is avoided, but also the memory access delay is reduced, therefore the throughput increases. In previous work, filters are stored inside the on-chip memories, but still, the temporary feature map generated in each layer is required to be written to and read from an external DRAM in run-time. In our work, however, both parameters and the temporary feature maps are stored in the on-chip memory.

3.4 Domain Specific Customization of ISA

BiNMAC consists of special instructions for Binary Neural Networks, such as XNOR, PCNT and PCH. All of these instructions require one clock cycle for the execution stage.

3.4.1 XNOR and Popcount Instruction. We added XNOR and PCNT to the ISA of the BiNMAC. XNOR and Population Count are the main operators in BNNs. Population Count operator counts the number of 1s in a data packet (16 bits). With a very small hardware overhead (10 full adders and 6 half adders) for a data packet, PCNT instruction can replace 4 lines of instruction assembly code, and 52 cycles in terms of execution, all fused in one instruction and one clock cycle. XNOR instruction is implemented with an array of 16 XNOR gates that simply performs the bit-wise XNOR operation between two registers or data memory entries.

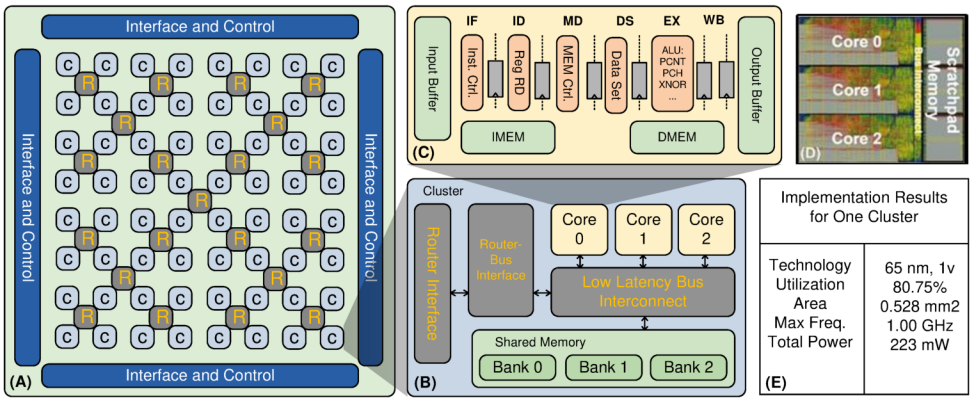

Figure 2: (A) BiNMAC comprising of 64 clusters (B) Cluster Architecture highlighting cluster memory, bus and processing cores (C) Processor architecture consisting of 6 pipeline stages. IF, ID, MD, DS, EX and WB are inst. fetch, inst. decode, memory decode, data set, execution, and write-back stages. IMEM and DMEM are instruction and local data memories. (D) Post-layout view of cluster implemented in 65nm, 1V TSMC CMOS technology (E) Post-layout implementation results of optimized bus-based cluster (consisting of 3 cores + bus + cluster memory).

3.4.2 Patch-select - PCH Instruction. PCH is a new invention to the ISA of BiNMAC which eases the bit packing and bit manipulation. Since coefficients are packed in 16 bit entries, there are times that a patch of a specific size is required to be read from one data entry, and to be shifted and written to another. PCHx is devised to meet this necessity. This optimization saves 4 lines of instruction and 4 clock cycles for the execution. PCHx in BiNMAC has been devised to handle patches of length 1 to 9. Fig. 3 shows an example of a PCH3 instruction, its hardware description language, and its function snippet that is frequently used in the process of a BNN.

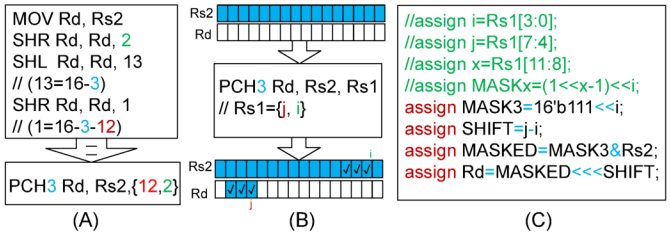

Figure 3: (A) A patch-select snippet and its equivalent instruction that fuses 4 lines of code and 4 execution clock cycles into one. (B) The register transfer illustration of the patch-select example (C) Verilog code that shows implementation of the patch-select instruction using barrel shifters

4 IMPLEMENTATION RESULTS AND PLATFORM COMPARISON

Both binarized 3-layer MLP and ResNet-20 are implemented on BiNMAC and the results are presented in terms of throughput, execution time and energy efficiency for standard and binarized neural networks. To better gauge the performance of BiNMAC, we also implemented the two binarized networks on NVIDIA Jetson TX2 SoC (CPU+GPU) platform. In BiNMAC implementation, all of BNN weights are distributed among the cluster. Due to the architecture scalability of BiNMAC, for different neural networks, different number of clusters are activated to accommodate the network parameters and temporary feature maps, and by means of its Dynamic Frequency Scaling (DFS), one can choose the best frequency given an application. The least number of required clusters, N, for a BNN is derived from equation (1) that guarantees that on-chip memories can always reserve the processing data:

$$N = \frac{NetworkParameterSize}{ClusterMemorySize - BiggestFeatureMapSize} \quad (1)$$

4.1 MLP for Physical Activity Monitoring

The implementation results for the 3-layer MLP are provided in table 1. For this binarized network, at least 4 clusters are required, 98% of their 12K word memory is used to accommodate the network parameters, and the rest accommodate feature map data. In the 3-layer MLP, by means of the DFS, the clock frequency is reduced to 145MHz, which is comparable to 140 MHz of the TX2 GPU. The implementation results for the binarized 3-layer MLP using 4 clusters with 12 cores are obtained in two different scenarios: Without using the PCH instruction It takes 0.085ms (at 145 MHz) to classify one input window of 40 sample whereas it takes 0.077ms if PCH is used at bit-packing phase, resulting in a performance improvement of 1.1× over the latter and 1.8× over the standard version of 3-layer MLP implemented on PENC with 53 clusters.

4.2 Binarized ResNet-20 for CIFAR-10

For the binarized ResNet-20 network, at least 16 clusters are required to be activated for processing and providing a memory of size 48K words, 66% of which is used for network parameters, and the rest is used for the feature map data. Table 1 shows the implementation results in terms of execution time, energy consumption, throughput, and energy efficiency. The implementation results for binarized ResNet-20 using the 16 clusters with 48 cores was done in two different scenarios: with and without PCH instruction. Without a PCH instruction, and only by using its equivalent function as in Fig. 6-A, it takes 1,865,441 clock cycles (1.86ms at 1 GHz) to classify one single image, resulting in a classification throughput of 537.6 label per second, and a power of 3.9 watts when running on 16 clusters at clock frequency of 1 GHz. On the other hand, when taking the PCH instruction into account, due to fusion of 4 instructions into one, which is frequently used in both patch selecting

Table 1: BiNMAC hardware implementation results for 3-layer MLP for the activity monitoring and Binarized ResNet-20 for CIFAR-10

Network/Platfrom	ResNet-20 (System Freq=1GHz)				3-layer MLP (System Freq=145MHz)			
	Standard	BiNMAC w/o PCH	BiNMAC w/ PCH	Improv. over Standard	Standard	BiNMAC w/o PCH	BiNMAC w/ PCH	Improv. over Standard
# of Used Clusters (N)	13	16	16	0.8x	53	4	4	13.2x
Execution Time (ms)	36	1.86	0.98	36x	0.139	0.085	0.077	1.8x
Power Consumption (W)	3.5	3.8	3.9	0.9x	1.9	0.27	0.27	7.3x
Energy Consumption (mJ)	116	7.3	3.8	30x	0.26	0.02	0.02	13x
Throughput (label/s)	27	537.6	1020	36x	7,194	11,765	12,987	1.8x
Throughput/Power (label/s/W)	7.9	137.4	260	30x	3,786	43,574	48,100	13x
EDP (mJs)	4.2	0.01	0.004	1,044x	3.6e-5	1.7e-6	1.54e-6	23x

Table 2: Comparison of BiNMAC w/ PCH results with NVIDIA TX2. For comparison the results are scaled to 28nm.

Network/Platform	ResNet-20				3-layer MLP			
	TX2	BiNMAC Serial	BiNMAC Fully Parallel	Improv. over TX2	TX2	BiNMAC Serial	BiNMAC Fully Parallel	Improv. over TX2
Tech (nm)	28	65	65	–	28	65	65	–
# of Used Clusters (N)	–	16	64	–	–	4	64	–
Frequency (MHz)	1033	1000	1000	–	140	145	145	–
Execution Time (ms)	21.8	0.98	0.28	78x	0.17	0.077	0.0076	22x
Throughput (label/s)	46	1020	3571	78x	5,882	12,987	131,579	22x
Power (W)	3.8	3.9	16.4	0.23x	2.1	0.27	4.84	0.43x
Scaled Power (W)	3.8	1.67	6.9	0.55x	2.1	0.12	2.1	1x
Scaled Energy Consumption (mJ)	83	1.63	1.94	41x	0.37	0.009	0.016	23x
Scaled Throughput/Power (label/s/watt)	12.8	608	512	41x	2800	108,496	63,013	23x

and bit packing phases, the number of clock cycles to classify one single image drops to 981,083 (0.98ms at 1 GHz), thus improving the throughput by almost 1.9×. The latter implementation on BiNMAC has a drastic performance improvement of 36× and an energy consumption per label improvement of 30× over those of a standard ResNet-20 implemented on the precedent version of BiNMAC.

4.3 Implementation Results on TX2 SoC

3-layer MLP and binarized ResNet-20 are implemented on NVIDIA TX2 platform. GPU clock frequency is 140 MHz and 1033 MHz for binarized 3-layer MLP and ResNet-20, respectively. Also, one CPU core is on, running at 345 MHz. Table 2 summarizes the results, and to make a fair comparison, the BiNMAC performance results are scaled to 28nm. Based on the results, BiNMAC achieves 22× and 78× higher throughput and 23× and 41× lower energy consumption compared to GPU for the 3-layer MLP and ResNet-20, respectively.

4.4 Comparison with Existing Work

Table 3 compares the BiNMAC results with state-of-the-art FPGA accelerators for CIFAR-10 dataset. BiNMAC has highest throughput compared to other work. Compared to [7] and [11], BiNMAC achieves 14× and 6× higher throughput. Also, BiNMAC consumes 1.2× lower power compared to [11] but 2× higher than [7].

Table 3: Comparison of BiNMAC with existing work

Design	**[7]**	**[11]**	**This work**
Platform	Artix-7(XC7A200T)	Zynq (7Z020)	Manycore
Latency (ms)	13	6	1
Throughput (img/sec)	72.6	168	1020
Frequency (MHz)	100	143	1000
Power (W)	1.82	4.7	3.9

5 CONCLUSION

We proposed an energy efficient Binarized neural Network Manycore ACcelerator named "BiNMAC" with specialized binary architecture, instructions, and memory access architecture. We also proposed a 3-layer MLP which is a binary neural network designed for multi-modal time-series data classification. Both fully-connected network and convolutional ResNet-20 were binarized and were deployed at BiNMAC. The results show BiNMAC consumes 0.02 mJ and 3.8 mJ energy which is 13× and 30× lower than standard non-binarized MLP and ResNet-20 deployed at an equivalent precedent platform. Furthermore, we implemented the two BNNs on the NVIDIA Jetson TX2 SoC (CPU+GPU), and BiNMAC achieves 22× and 78× higher throughput and 23× and 41× lower energy consumption for binarized 3-layer MLP and ResNet-20, respectively.

REFERENCES

[1] Matthieu Courbariaux et al. 2016. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830* (2016).

[2] Kaiming He et al. [n. d.]. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[3] Ali Jafari et al. 2017. An embedded FPGA accelerator for a stand-alone dual-mode assistive device. *Memory (Kb)* 101 (2017), 102.

[4] Ali Jafari et al. 2018. A Low-Power Wearable Stand-Alone Tongue Drive System for People With Severe Disabilities. *IEEE transactions on biomedical circuits and systems* 12, 1 (2018), 58–67.

[5] Alex Krizhevsky et al. [n. d.]. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*.

[6] Alex Krizhevsky and Geoffrey Hinton. 2009. Learning multiple layers of features from tiny images. (2009).

[7] Adam Page et al. 2017. Sparcnet: A hardware accelerator for efficient deployment of sparse convolutional networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 13, 3 (2017), 31.

[8] Mohammad Rastegari et al. 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*. Springer, 525–542.

[9] Attila Reiss and Didier Stricker. 2012. Introducing a new benchmarked dataset for activity monitoring. In *Wearable Computers (ISWC), 2012 16th International Symposium on*. IEEE, 108–109.

[10] Yaman Umuroglu et al. 2017. Finn: A framework for fast, scalable binarized neural network inference. In *Proceedings of the SIGDA*. ACM.

[11] Ritchie Zhao et al. 2017. Accelerating Binarized Convolutional Neural Networks with Software-Programmable FPGAs.. In *FPGA*. 15–24.