A Reconfigurable Inference Processor for Recurrent Neural Networks Based on Programmable Data Format in a Resource-Limited FPGA

Jiho Kim, Kwoanyoung Park, and Tae-Hwan Kim
School of Electronics and Information Engineering, Korea Aerospace University
76, Hanggongdaehak-ro, Deogyang-gu, Goyang-si, Gyeonggi-do, 10540, Republic of Korea
e-mail : taehwan.kim@kau.kr

Abstract— An efficient inference processor for recurrent neural networks is designed and implemented in an FPGA. The proposed processor is designed to be reconfigurable for various models and perform every vector operation consistently utilizing a single array of multiply-accumulate units with the aim of achieving a high resource efficiency. The data format is programmable per operand. The resource and energy efficiency are 1.89MOP/LUT and 263.95GOP/J, respectively, in Intel Cyclone-V FPGA. The functionality has been verified successfully under a fully-integrated inference system.

I. Introduction

There are various recurrent neural network (RNN) models such as the long short-term memory (LSTM) and gated-recurrent unit (GRU), which can fulfil the stateful inference. It is well known [1, 2] that no unique model absolutely outperforms the others in every of the design aspects including the workload and inference performance. Hence, various models thus need to be supported by an RNN inference processor if the processor targets diverse applications that have different design objectives and constraints from each other. In addition, its resource efficiency is important when implemented in a resource-limited FPGA.

This paper presents an efficient RNN inference processor. The proposed processor is reconfigurable to support various models, as designed to be an instruction-set processor to perform the primitive vector operations commonly involved in the models. The proposed processor utilizes a single array of multiply-accumulate (MAC) units efficiently to perform every vector operation achieving a high resource efficiency. In addition, the proposed processor provides the programmability of the data format for each vector operand. Implemented in Intel Cyclone-V FPGA, the proposed processor shows the resource and energy efficiency of 1.89MOP/s/LUT and 263.95GOP/J, respectively. The functionality has been verified for various models under an inference system developed by integrating the proposed processor.

II. Proposed Processor

The proposed processor has been designed to be a SIMD instruction-set processor to perform the vector operations efficiently. Fig. 1 shows the microarchitecture along with its supported vector instructions. The dataflows of the RNN models have the three primitive vector operations in common [2]: matrix-vector MAC (MVMA), elementwise MAC (EMAC), and elementwise non-linear function (ENOF), each of which can be performed by the corresponding instruction in Fig. 1. The processor performs 64 scalar operations per cycle through the six pipeline stages with the operands stored in the memories. ACU queries the tables with the activation data for the coefficients, which are going to be used to evaluate the non-linear functions by employing the linear splines. VPU has been designed by incorporating an array of MAC units. Every vector operation is performed consistently by the single array of MAC units with a high utilization rate, leading to a high resource efficiency.

In the proposed processor, the data format of each vec-

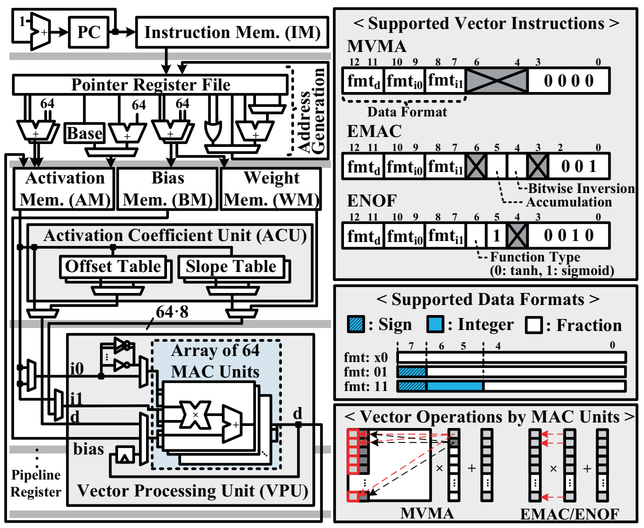

Fig. 1. Proposed processor.

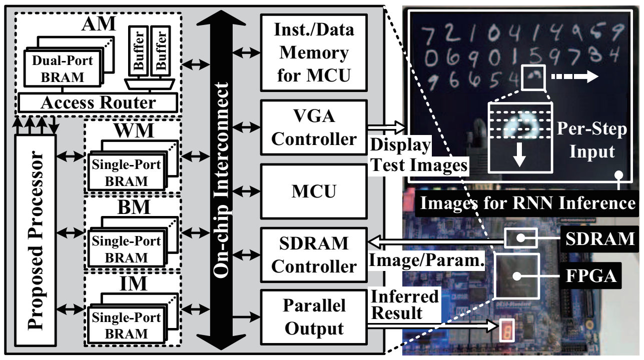

Fig. 2. Overall inference system.

tor operand is programmable, where the supported data formats are shown in Fig. 1. It is not efficient to support only a single format considering the different data distributions of the vectors; for example, the data resulting from the sigmoid function are positive and not greater than 1 while the weight or activation data may have a different distribution. In the proposed processor, the data format for each vector operand can be programmed to what is adequate to represent its range and/or precision. It is noteworthy that the data size in the proposed processor is much smaller than those in the previous processors [2, 3] while the inference results are maintained.

A prototype inference system has been developed by integrating all the essential components in order to verify the functionality of the proposed processor. Fig. 2 shows the overall architecture of the inference system. In the system, the memories that are associated directly with the proposed processor, i.e., AM, WM, BM, and IM, were designed by instantiating BRAMs. To realize a high bandwidth that is required by the proposed processor working without the pipeline stalls, the memories are designed based on the multi-bank structures. The inference procedure is actualized as follows: MCU preloads the data of the dataflow description program, weight, and bias into IM, WM, and BM, respectively from the external SDRAM; for each step, MCU provides the input data to the proposed processor and the proposed processor performs the inference by running the program.

III. Results and Conclusion

The implementation results are summarized in Table I. Fitted in Intel Cyclone-V FPGA, which is a low-cost resource-limited device, the proposed processor shows the peak inference speed of 22.66GOP/s at 118MHz with 1.1V supply and 85°C. The resource usage of the proposed processor is much lower than the previous results, thanks to the efficient architecture based on a single array of MAC units as well as the programmable data format. The resource efficiency of the proposed processor is 4.1 times higher than the previous state-of-the-art result. The functionality of the proposed processor has been verified for several different models and the results obtained from some of them are summarized in Table II, where the inference performances for the sequential MNIST and word-level Penn Treebank tasks [4] have been evaluated in terms of the classification accuracy and perplexity (ppl), respectively. The state size of the models has been configured to 128. The performance loss has been obtained by the difference from the inference performance achieved with the 32-bit floating-point data. The proposed processor achieves a near-floating-point performance with the 8-bit data whose formats are programmable. The demonstration video is accessible via `https://youtu.be/ni7mCzl-BcE`.

TABLE I
FPGA implementation results.

Processor	Proposed	[2]	[3]
FPGA device	Cyclone V	Cyclone V	Virtex-7
Model reconfigurability	Yes	Yes	No (only LSTM)
Data format	Programmable 8bit	Fixed 16bit	Fixed 16bit
Inf. speed (GOP/s)	22.66	23.00	10.90
LUT resource usage	12K	18K	23K
BRAM res. usage (bit)	801K	1620K	5940K
Res. eff. (MOP/s/LUT)	1.89	1.28	0.46
Energy eff. (GOP/J)	263.95	166.31	7.10

TABLE II
Inference performance of the proposed processor.

Task	Sequential MNIST		Word-level Penn Treebank	
RNN model	LSTM	Peephole GRU	Peephole LSTM	Bidirect. GRU
Inf. performance	98.65%	98.17%	95.17 (ppl)	2.74 (ppl)
Performance loss	0.22%	0.61%	0.73 (ppl)	0.98 (ppl)
Per-step latency	13.64μs	8.56μs	11.95μs	20.51μs

Acknowledgements

This work was supported by IITP grant funded by the Korea government (MSIT) [2017-0-00528]. The EDA tools were supported by the IC Design Education Center.

References

[1] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv:1412.3555, pp.1-9, Dec. 2014.

[2] J. Kim, J. Kim, and T.-H. Kim, "AERO: a 1.28 MOP/s/LUT reconfigurable inference processor for recurrent neural networks in a resource-limited FPGA," *MDPI Elect.*, vol. 10, no. 11, pp. 1249-1263, Apr. 2021.

[3] W. Zhang, F. Ge, C. Cui, Y. Yang, F. Zhou, and N. Wu, "Design and implementation of LSTM accelerator based on FPGA," in *Proc. Int'l Conf. Comm. Tech.*, IEEE, Oct. 2020, pp. 1675-1679.

[4] A. Ardakani, Z. Ji, and W. J. Gross, "Learning to skip ineffectual recurrent computations in LSTMs," in *Proc. Design, Auto. & Test in Europe Conf. & Exhib.*, IEEE, Mar. 2019, pp. 1427-1432.