**Seyedramin Rasoulinezhad**
*Camperdown NSW 2006, Australia*
*+1 780 695 1008*
✉ *seyedramin.rasoulinezhad@sydney.edu.au*

31$^{th}$ August 2021

To the TRETS Editors,

Please find attached, a revision to Manuscript ID TRETS-2021-0058 entitled "Rethinking Embedded Blocks for Machine Learning Applications", which was previously submitted to Transactions on Reconfigurable Technology and Systems (TRETS) for consideration in the FPT 2021 Journal Track. Also, a full response letter alongside the revised manuscript is provided. All modifications to the manuscript are highlighted in red.

We confirm that this work is original and the submitted content has not been published elsewhere, nor is it currently under consideration for publication elsewhere.

This work points the inefficiencies in the FPGA embedded blocks considering the new demand for deployment of embedded machine learning algorithms, where current architectures are excessively optimized for high-precision arithmetic. With the goal of exploring this new design space in a methodical manner, we first propose a problem formulation involving computing nested loops over multiply-accumulate (MAC) operations, which covers many basic linear algebra primitives and standard deep neural network (DNN) kernels. A quantitative methodology for deriving efficient coarse-grained compute block architectures from benchmarks is then proposed together with a family of new embedded blocks, called MLBlocks. Using synthetic benchmarks, we demonstrate that MLBlocks offer significantly improved performance over the commercial Xilinx DSP48E2 architecture, while maintaining similar area and delay.

We believe that this manuscript is appropriate for publication by TRETS because it is an architectural exploration for reconfigurable embedded blocks for FPGA technology. This perfectly matches the TRETS aims and research scope.

As we plan to make our framework open source, we also request the artifact evaluation process. The main requirements to replicate our results are 1) a Linux machine, 2) Encounter(R) RTL Compiler RC14.11, and 3) STMicro 28nm technology library files.

Thank you for your consideration of this manuscript.

Sincerely,

Seyedramin Rasoulinezhad