Analysis of Various Approximate Multipliers

Jinay Dagli, Krishil Gandhi, and Neel Shah

Mentor: Mr. Kailash Prasad

Supervisor: Prof. Joycee Mekie

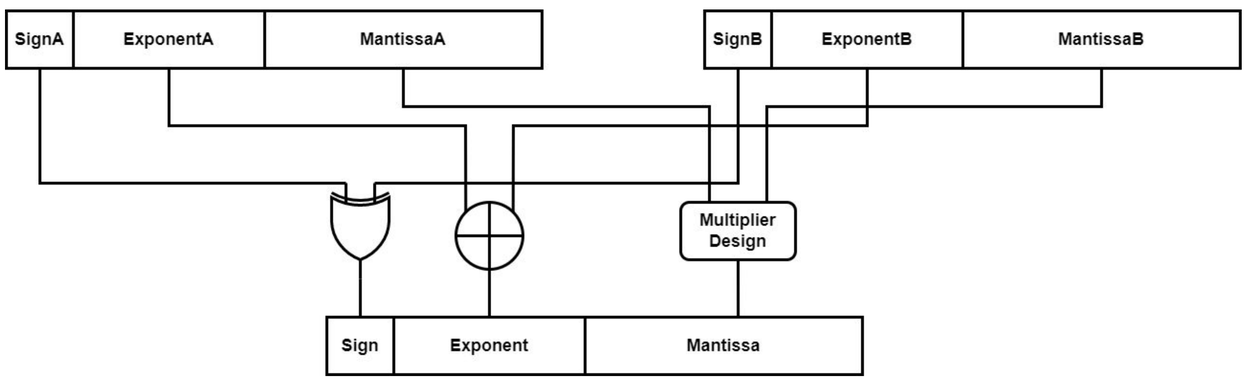
Indian Institute of Technology Gandhinagar

INTRODUCTION

The IEEE 754 standard has been used in various applications over the years. We used two techniques (the newly developed quantisation based on POSIT number system, and approximation) that could lead to a reduction in the hardware, power, and area requirements of the multiplier, while maintaining the accuracy for several Neural Networks, mainly focussing on LeNet5 and ResNet18. We have implemented various multipliers: Fixed-POSIT multiplier, approximate integer, approximate floating-point multipliers (Compressor-based, FPCAM, RMAC, CFPU), along with various approximate multipliers proposed over the years.

METHODOLOGY

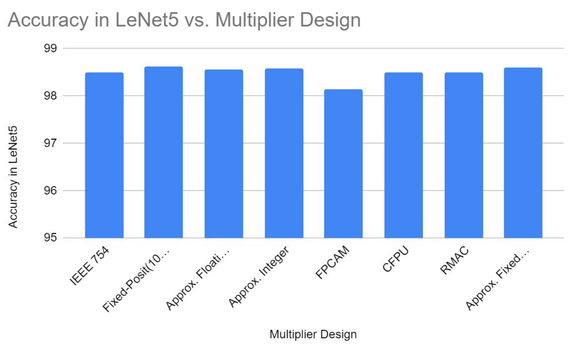
The general flow of the setup is as follows:

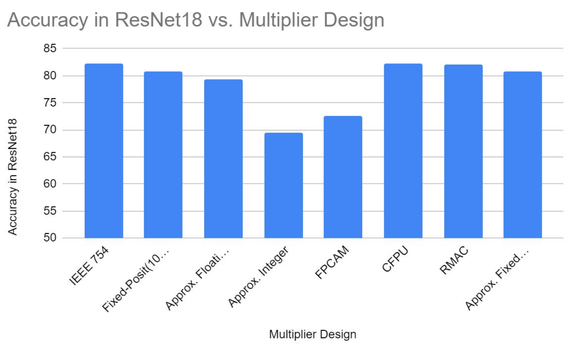


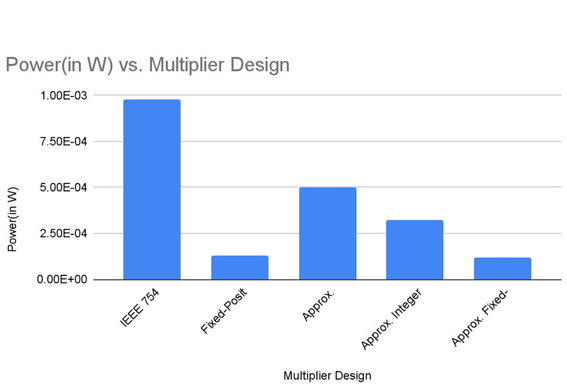
The multiplier design is replaced by one of the various multipliers we have tested.

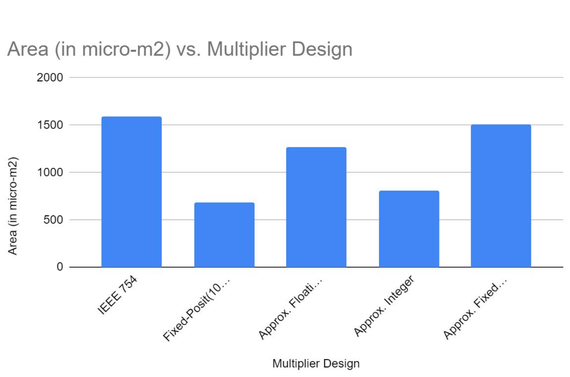
RESULTS

The following results for two Neural Networks have been obtained:









Note that:

1. For compressor-based approximate design, we have plotted the accuracy, power and area, when using the Yang2 [1] compressor.

2. For power and area comparison, the IEEE 754 standard, the approximate fixed-posit, and approximate floating-point multipliers use a total length of 16 bits.

CONCLUSIONS

- As could be observed from the plots in the Results section, it is evident that the approximate integer multiplier can be used only for smaller networks, but it accuracy drops as we go on to larger networks.

- Similarly, the compressor-based approximate floating-point multiplier works well for 32 bits, but as we decrease the number of bits of mantissa, accuracy would degrade.

- FPCAM (Floating Point Configurable Approximate Multiplier), as observed, consumes very less area and power, but we have to trade accuracy in larger networks.

- Similarly, CFPU (Configurable Floating-Point) and RMAC (Runtime Configurable) multiplier both have quite less drop in the accuracy as compared to the base accuracy.

- An another important observation is the difference in the accuracies on changing the operand sequence in the multiplier.
  That is, multiplying *a with b* and multiplying *b with a* would give us different accuracies in case of approximate multipliers.[6]

- For the compressor-based approximate multipliers (approximate integer, approximate floating-point, and approximate fixed-posit multipliers), it is observed that the power and area consumption is much less than the standard IEEE 754 multiplier.

  Amongst the compressor designs [1], Yang2 is observed to be the most accurate design (although consuming more area and power as compared to other compressor designs). Similarly, the Sabetz multiplier is observed to have the least accuracy, but is quite efficient in terms of area and power. Note that for the results, best case accuracy is reported (accuracy of Yang2).

- A final observation that was made is that when the number of mantissa bits are eight or more, we could use the compressor-based approximation technique for hardware reduction. Else, we could use the techniques used in other approximate multipliers, such as FPCAM, RMAC, or CFPU.

REFERENCES

1. Antonio Giuseppe Strollo, Ettore Napoli, Davide De Caro, et al. "Comparison and extension of approximate 4-2 compressors for low-power approximate multipliers". In: IEEE Transactions on Circuits and Systems I: Regular Papers 67.9 (2020), pp. 3021–3034.
2. C. K. Jha, S. Walia, G. Kanojia and J. Mekie, "FPCAM: Floating Point Configurable Approximate Multiplier for Error Resilient Applications," 2021 IEEE International Symposium on Circuits and Systems (ISCAS), 2021, pp. 1-5, doi: 10.1109/ISCAS51556.2021.9401341.
3. M. Imani, D. Peroni and T. Rosing, "CFPU: Configurable floating point multiplier for energy-efficient computing," 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC), 2017, pp. 1-6, doi: 10.1145/3061639.3062210.
4. Mohsen Imani, Ricardo Garcia, Saransh Gupta, and Tajana Rosing. 2018. RMAC: Runtime Configurable Floating Point Multiplier for Approximate Computing. In Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED '18). Association for Computing Machinery, New York, NY, USA, Article 12, 1–6. https://doi.org/10.1145/3218603.3218621
5. Gohil, Varun, et al. "Fixed-Posit: A Floating-Point Representation for Error-Resilient Applications." *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 10, 2021, pp. 3341–3345., https://doi.org/10.1109/tcsii.2021.3072217.
6. A. Nandi, C. K. Jha and J. Mekie, "Should We Code Differently When Using Approximate Circuits?," 2019 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2019, pp. 37-40, doi: 10.1109/APCCAS47518.2019.8953113.

ACKNOWLEDGEMENTS

We would like to thank Mr. Kailash Prasad for his guidance throughout the course of this project. We would also like to thank Prof. Joycee Mekie for some wonderful ideas throughout the project and for giving us an opportunity to pursue this project.