HBMorphic: FHE Acceleration via HBM-Enabled Recursive Karatsuba Multiplier on FPGA

Hassan Nassar*, Lars Bauer, Jörg Henkel*
*Chair for Embedded Systems, Karlsruhe Institute of Technology, Germany hassan.nassar@kit.edu

***Abstract*—Cloud computing offers advantages such as seamless scalability and speedup of computation. Nevertheless, these benefits come with notable tradeoffs, e.g., processing sensitive data without compromising security. Fully Homomorphic Encryption (FHE) solves this by processing of encrypted data. In this work, we develop an FHE hardware accelerator that uses a custom control interface to maximally utilize the bandwidth of HBM, following the memory access patterns of FHE.**

I. Introduction

Cloud computing allows the usage of on-demand services. However, this faces concerns over privacy and security [1]. In response to these concerns, Homomorphic Encryption (HE) emerges as a robust solution, it allows processing over encrypted data. HE ensures that sensitive information remains confidential throughout computational processes. HE applications span diverse domains [2]. However, HE has a substantial computational and memory overhead of homomorphic operations. On the algorithmic side, mathematical optimizations exist. The most prominent of them is Fully Homomorphic Encryption over the Torus (TFHE), which is post-quantum secure [3]. In this work, we implement an accurate accelerator for TFHE on an HBM-enabled FPGA.

II. HBMorphic's Design & Implementation

We implement our accelerator on a VCU128 board [4]. The HBM has 32 Pseudo Channels (PCs), each of size 256 MiB. The chip includes a generic interface containing an ASIC interconnect between the PCs and the FPGA. It can be used as it is or bypassed by implementing a custom interface to get higher throughput [5]. Our design would work the same on other FPGA or FPGA boards, that include HBM with a similar specification.

We implement our own HBM interface to achieve the highest throughput possible. For one external product for PBS of TFHE-777, a total of 25 MiB is read. The data needed for the external product is packed in 777 3D arrays each of the size $\{4, 4, 512\}$ of 32 bit words. We divide the 32 PCs over the 3D arrays equally to have as least memory contentions as possible. Each PC is used to only read 256 words, distributing the load symmetrically.

To access each of the PCs independently, we create 32 AXI memory interfaces, and each is capable of managing the read and write requests for one PC. Our Karatsuba-based accelerator is contained in an AXI wrapper with 32 independent ports. Each PC has a data output width of 64 bits.

This work was partially funded by the "Helmholtz Pilot Program for Core Informatics (kikit)" at KIT and the German Federal Ministry of Education and Research (BMBF) through the Software Campus Project.

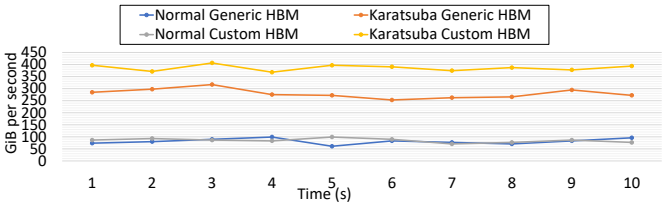

Fig. 1: Maximum Bandwidth utilization

The frequency of the HBM is higher than the frequency of the logic implemented on the FPGA. Therefore, for each port, we use a bitwidth of 512 bit packing four words from each PC together. The 256 words are read sequentially over 16 read operations. To amortize the latency as much as possible, we use double buffering. Therefore, the data is already available immediately when it is needed.

We track our accelerator's memory access pattern of the data for 10 seconds using the HBM monitor from Xilinx [4]. The theoretical maximum bandwidth of the HBM on the VCU128 board is 460 GiB/s. However, based on the Xilinx documentation, practically the limit is 90% of this theoretical bandwidth [5]. For the normal multiplier, it did not make a difference between the custom and generic interface. In general, it never broke the limit of 100 GiB/s. Using HBMorphic leads to the highest bandwidth utilization as Fig. 1 shows. It reaches a maximum of 406 GiB/s. The highest the Karatsuba multiplier achieved using the generic interface was 316 GiB/s. HBMorphic reached 88% of the theoretical bandwidth which is very near to the 90% mark that Xilinx mentions as the practical maximum bandwidth.

III. Conclusion

In this work, we introduce a fully homomorphic encryption accelerator on an FPGA with HBM. HBMorphic accelerates the state-of-the-art TFHE algorithm and loads the data with high throughput via our custom HBM interface reaching 88% of the HBM bandwidth.

References

[1] Z. Xiao *et al.*, "Security and privacy in cloud computing", *IEEE Communications Surveys & Tutorials*, 2013.
[2] K. B. Johnson *et al.*, "Precision medicine, AI, and the future of personalized health care", *Clinical Translational Science*, 2020.
[3] I. Chillotti *et al.*, "TFHE: Fast fully homomorphic encryption over the torus", Cryptology ePrint Archive, Paper 2018/421, 2018.
[4] *VCU128 Board User Guide UG1302*, Xilinx, Inc., 2022.
[5] *Vitis Tutorials: Hardware Acceleration XD099*, Xilinx, Inc., 2023.