High-Throughput FPGA Acceleration of the Number Theoretic Transform for FV Homomorphic Encryption

Benjamin Steeg, Zahra Mojtahedin, Hassan Nassar, Jörg Henkel

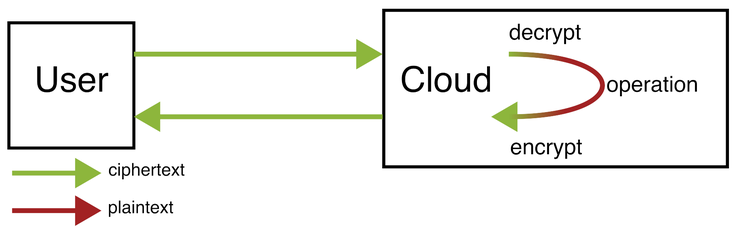

Figure: without Homomorphic Encryption

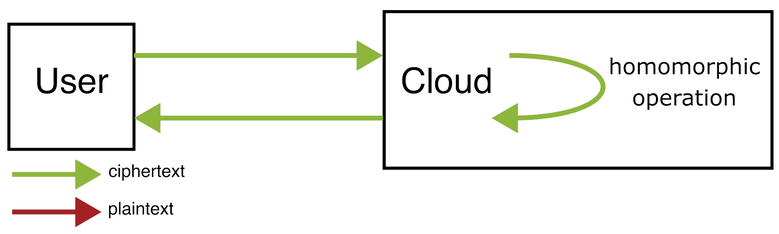

Figure: with Homomorphic Encryption

Basics

- Homomorphic Encryption (HE): Operations can be performed on ciphertexts without prior decryption
- Useful for cloud computing on sensitive data
- Potential use cases:
  - Machine Learning: inference and training of NNs
  - Analysis on medical data
  - Electronic voting systems
- Generally computationally intensive
- Our Choice: FV encryption scheme
  - Plaintexts: Polynomials
  - Addition, Multiplication
- Polynomial multiplication is largest part of computation

Number Theoretic Transform

- DFT variant over a finite field
- Can use FFT algorithms
- Used in most accelerators
- Multiply polynomials $a, b \in \mathbb{Z}_q[X]/(X^n + 1)$ via:

$$a \cdot b = \mathrm{INTT}(\mathrm{NTT}(a) \circ \mathrm{NTT}(b))$$
$$\text{where } \circ\text{: coefficient-wise multiplication}$$

- Polynomial multiplication in $O(n \log n)$ instead of $O(n^2)$
- Without NTT: >100m int multiplications per homomorphic multiplication
- With NTT: <500k int multiplications per homomorphic multiplication

Approach

- Parameters
  - Length of polynomials: $n = 4096$
  - Width of primes and coefficients: 30 bit
- Two coefficients combined to one 60 bit "packed coefficient"
- Top level architecture is direct implementation of formula for polynomial multiplication:
  - 2 NTT processors
  - Set of multipliers
  - 1 INTT processor
  - RAM (used for buffering; not needed if outer logic can accept output fast enough)
  - Control logic
- 2-Stage coarse grained pipeline:
  - 1. Stage: NTT + multipliers
  - 2. Stage: INTT
  - Because of one additional multiplication in the INTT, both stages take the same time.
- NTT Processor:
  - Compues NTT of one input polynomial
  - Iterative algorithm
  - Consists of C cores, where $C > 1$ and C is a power of 2.
  - During output: Processor passes $4C$ coefficients per cycle.
  - Control implements NTT algorithm as FSM and passes parameters to cores and router.

Our Goal

Develop an FPGA accelerator for NTT-based polynomial multiplication for use in FV homomorphic encryption.

- NTT Core:
  - Perform basic operations: CT butterflies
  - Fine grained pipeline for modular arithmetic
  - Modular reduction via sliding window with lookup table
  - RAM for coefficients
  - ROM for twiddle factors (needed in CT butterflies)
- Router:
  - Maps output from cores to correct core for next iteration
  - Implements custom assignment strategy to prevent empty cycles in cores
  - Outputs final result of computation
- INTT Processor: Structurally equivalent to NTT processor, however
  - INTT Cores (equivalent to NTT cores except GS butterflies instead of CT butterflies)
  - Additional multiplication needed at end of INTT
  - Control logic implements INTT

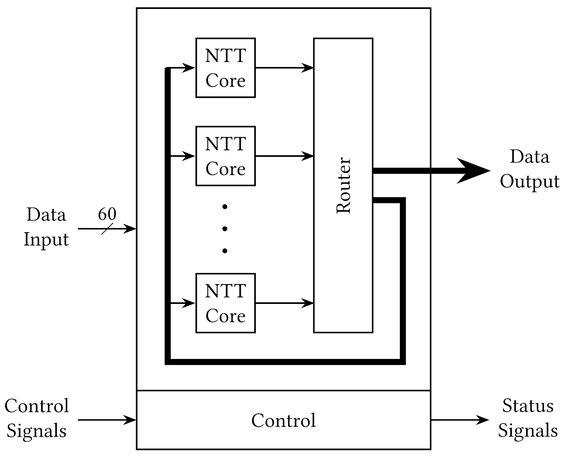

Figure: NTT Processor Architecture

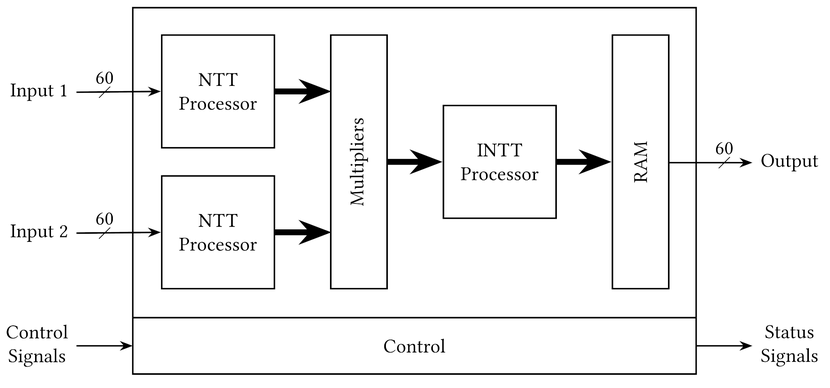

Figure: Top Level Processor Architecture

Results

- Implemented on AMD UltraScale+
- 3 versions: 8 cores, 4 cores, 2 cores (per processor)
- $f = 270\mathrm{MHz}$
- 4 core $f_{\max} = 279\mathrm{MHz}$, 2 core $f_{\max} = 283\mathrm{MHz}$
- Measured execution time (t_e) for single polynomial multiplication
- Measured computations per second (CPS) in pipelined operation
- Comparison against software on:
  - Apple M1
  - Intel i9-12900
  - Intel i7-12500
  - AMD Ryzen 7 2700X

- For single execution time:
  - 8C outperforms fastest CPU by 3.6x
  - 2C outperforms fastest CPU by 1.7x
- For pipelined operation:
  - 8C outperforms fastest CPU by 6.9x
  - 2C outperforms fastest CPU by 3.1x
  - Significantly increases needed memory bandwidth
- 8C (4C) is 2x faster than 4C (2C) when compared on computation time
- Memory I/O reduces effective speedup between versions
- Resource consumption roughly doubles between versions

C	f (MHz)	LUT	FF	BRAM	DSP	Power (W)
8	270	172407	31938	112	448	9.012
4	270	85443	17403	56	224	4.133
4	279	85411	19045	56	224	4.378
2	270	45437	9810	24	112	2.275
2	283	45531	10249	24	112	2.413

Table: Resource Consumption

Source Code

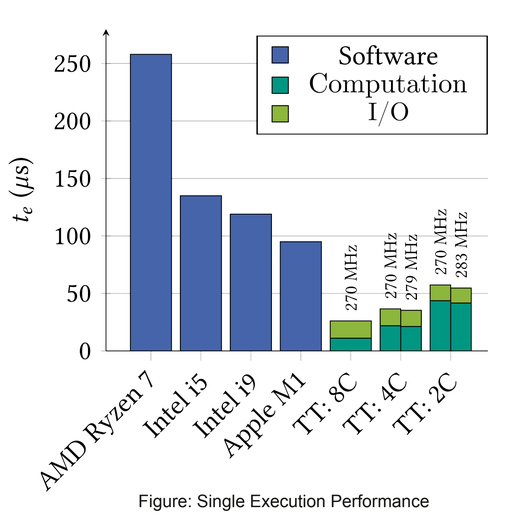

Figure: Single Execution Performance

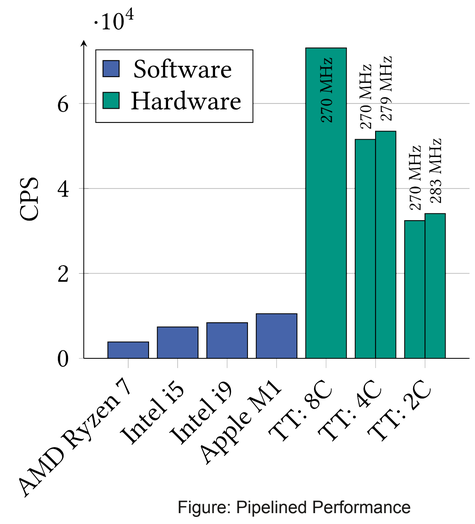

Figure: Pipelined Performance

Selected Literature

- J. Fan and F. Vercauteren, Somewhat practical fully homomorphic encryption, Cryptology ePrint Archive, Paper 2012/144, 2012.
- Zhen Liang and Yexin Zhao. 2022. Number Theoretic Transform and Its Applications in Lattice-based Cryptosystems: A Survey. doi:10.48550/arXiv.2211.13546
- Sujoy Sinha Roy, Furkan Turan, Kimmo Järvinen, Frederik Vercauteren, and Ingrid Verbauwhede. 2019. FPGA-Based High-Performance Parallel Architecture for Homomorphic Computing on Encrypted Data. In 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA). 387–398. doi:10.1109/HPCA.2019.00052
- A. C. Mert, E. Karabulut, E. Öztürk, E. Savaş, and A. Aysu. 2022. An Extensive Study of Flexible Design Methods for the Number Theoretic Transform. IEEE Trans. Comput. 71, 11 (Nov. 2022), 2829–2843. doi:10.1109/TC.2020.3017930