University of California
Santa Barbara

Architectural Support and Modeling of Emerging Technologies for Datacenter Privacy and Security Applications

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Electrical and Computer Engineering

by

Alvin Oliver Glova

Committee in charge:

Professor Timothy Sherwood, Chair
Professor Jonathan Balkind
Professor Dmitri Strukov
Professor Tao Yang

September 2022

The Dissertation of Alvin Oliver Glova is approved.

Professor Jonathan Balkind

Professor Dmitri Strukov

Professor Tao Yang

Professor Timothy Sherwood, Committee Chair

September 2022

Architectural Support and Modeling of Emerging Technologies for Datacenter Privacy and Security Applications

by

Alvin Oliver Glova

Dedicated in memory of my parents,
Alvin and Violeta

Acknowledgements

I want to express my utmost gratitude to my advisor, Tim Sherwood. I'm grateful for the countless hours of discussions on research, productivity, career path, everyday life, among other things. I have learned a lot from him as he truly excels as an academic and life advisor. Also, thank you to the other faculty members of my dissertation and qualifying exam committees, Dmitri Strukov, Tao Yang, Yuan Xie, and Jonathan Balkind for all the constructive feedback I have received. Special thanks to Jonathan who I have collaborated with since he joined the CS Department and has provided valuable assistance on many projects since. To Yuan and members of the SEAL Lab, where I spent my first few years in my PhD, thank you for welcoming me and guiding me in those years.

I would also like to extend my warmest thanks to the following:

To all my colleagues and friends I met or worked with in UCSB ArchLab, SEAL Lab, ECE, CS Departments and my internships at Berkeley Lab and Samsung. Special thanks to Chris, Jeremy, Deeksha, Rhys, George, Jennifer, Jim, Alexis of ArchLab for all the collaboration and fun memories in the lab. To Itir, Shuangchen, Abanti, Peng, Maohua, Xinfeng, Dylan, Liang, and Xing of SEAL Lab for the assistance and great memories together. To friends I met in grad school, hiking trips, and badminton games: Nima, Animesh, Nazerke, Vasilis, Changyun, Hanbyeol, Wayne, Peter, Jonathan, Suranjan, Jason, It, and Sean. Thank you for all the great times we had together and making grad school a lot more fun. To my siblings, thank you for the everlasting support. To my parents, who would have been happy to see me achieve this milestone in my life.

Finally, to the love of my life, Gowoon, for the continuous support and encouragement from the very beginning of this journey.

Curriculum Vitæ

Alvin Oliver Glova

Education

2022	Ph.D. in Electrical and Computer Engineering, University of California, Santa Barbara.
2012	M.S. in Electrical Engineering, Korea Advanced Institute of Science and Technology.
2009	B.S. in Computer Engineering, University of the Philippines, Diliman.

Selected Professional Experience

2019	Research Intern, Lawrence Berkeley National Lab, Berkeley, CA.
2017	Research Intern, Samsung Semiconductor Inc., San Jose, CA.
2012 - 2016	Research Engineer, SK Hynix, Icheon, South Korea.

Publications

- **Alvin Oliver Glova**, Yukai Yang, Yiyao Wan, Zhizhou Zhang, George Michelogiannakis, Jonathan Balkind, Timothy Sherwood, "*Establishing Cooperative Computation with Hardware Embassies*", IEEE International Symposium on Secure and Private Execution Environment Design (SEED), Sept. 2022

- **Alvin Oliver Glova**, Jeremy Lau, Jonathan Balkind, Timothy Sherwood, "*Modeling a Hybrid Superconducting Processor*", In Submission

- Zhizhou Zhang, **Alvin Oliver Glova**, Jonathan Balkind, Timothy Sherwood, "*A Prediction System Service*", In Submission

- Deeksha Dangwal, **Alvin Oliver Glova**, Abhejit Rajagopal, Rhys Gretsch, Pranjali Jain, Jonathan Balkind, Timothy Sherwood, "*A Privacy-Optimizing Streaming Data Architecture*", In Preparation

- Jinjin Shao, Shiyu Ji, **Alvin Oliver Glova**, Yifan Qiao, Timothy Sherwood, Tao Yang, "*Index Obfuscation for Oblivious Document Retrieval in a Trusted Execution Environment*", ACM International Conference on Information and Knowledge Management (CIKM), Oct. 2020

- **Alvin Oliver Glova**, Itir Akgun, Shuangchen Li, Xing Hu, Yuan Xie, "*Near-Data Acceleration of Privacy-Preserving Biomarker Search with 3D-Stacked Memory*", IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE), Mar. 2019

- Shuangchen Li, **Alvin Oliver Glova**, Xing Hu, Peng Gu, Dimin Niu, Krishna T. Malladi, Hongzhong Zheng, Bob Brenan, Yuan Xie, "*SCOPE: A Stochastic Computing Engine for DRAM-based In-situ Accelerator*", IEEE/ACM International Symposium on Microarchitecture (MICRO), Oct. 2018

- Mimi Xie, Shuangchen Li, **Alvin Oliver Glova**, Jingtong Hu, Yuan Xie, "*Securing Emerging Nonvolatile Main Memory with Fast and Energy-Efficient AES In-Memory Implementation*", IEEE Transactions on Very Large Scale Integration Systems (TVLSI), Sept. 2018

- Mimi Xie, Shuangchen Li, **Alvin Oliver Glova**, Jingtong Hu, Yuangang Wang, Yuan Xie, "*AIM: Fast and Energy-efficient AES in-memory Implementation for Emerging Non-volatile Main Memory*", IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE), Mar. 2018

- Liang Chang, Zhaohao Wang, **Alvin Oliver Glova**, Jishen Zhao, Youguang Zhang, Yuan Xie, Weisheng Zhao, "*PRESCOTT: Preset-based Cross-point Architecture for Spin-orbit-torque Magnetic Random Access Memory*", IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Nov. 2017

Abstract

Architectural Support and Modeling of Emerging Technologies for Datacenter Privacy and Security Applications

by

Alvin Oliver Glova

As computing continues to be used for increasingly private and sensitive operations impacting all aspects of our lives, the need to maintain tight control of those computations only continues to grow. This, when coupled with the increasing trend of "outsourced" computation where datacenters are responsible for both storing data and performing computations over it on behalf of another party, naturally raises the level of importance of security and privacy even further. As such, algorithmic approaches to privacy-preserving and secure/trusted computations are rapidly emerging as a key aspect of workloads in datacenters at all scales. The higher cost associated with this additional algorithmic complexity will only increase the power consumption of these data centers, which are already receiving significant scrutiny for their ever more power-intensive operation. Architectural solutions are needed to support these emerging aspect of workloads. With the decline of Moore's Law, this also presents an interesting prospect for several energy-efficient "Post-Moore" technologies such superconducting electronics and steep-slope devices which are studied and developed as potential replacements for Silicon-based CMOS to realize low power datacenter processors and accelerators.

In this dissertation, we study new opportunities for architectural support of these emerging application needs in both traditional and emerging technologies. To perform this work we need to make additional contributions advancing the modeling and evaluation of emerging Post-Moore technologies in the context of secure privacy-preserving computations. First, we show how using a small, co-located, trusted hardware device can be used to improve multiparty computation-based operations based on the trade-off of physical security and performance. Second, we show how near-data processing can be exploited to improve certain forms of homomorphic encryption with applications in private search. Finally, we explore how these emerging technologies can be used to improve energy-efficiency of datacenter workloads by modeling accelerators and multicore processors.

Contents

Chapter 1

Introduction

We are living in a world where more and more data is being generated. Whether it is personal data from smartphones and smartwatches or big data records from enterprises, our systems are currently inundated with data. We can gain useful knowledge and insights from processing this data but, more often times than not, the only way to do this requires sharing of private data to remote servers in datacenters. These datacenters offer significant advantages in terms of availability, efficiency, scalability and often exclusive data resources such as models and records required for processing of client data. For third-party datacenters, operations at these scales provide far more than cost savings and convenience: they are continually monitored for failure, have significant and redundant connectivity to the outside world, are carefully managed 24/7, and are consistently upgraded.

However, there is a threat that malicious datacenter operators can get hold of private information and the servers themselves can also be potentially compromised and allow further leakage of information outside of the datacenter. This sharing then begs the question, how do pass trust to remote machines in datacenters? Is there a way we can operate on the data as if it is local in terms of security? Clearly there is a strong need to protect private information for data access, processing, and analysis on datacenters.

Because of this, in recent years, there has been a strong push for techniques for secure outsourcing of computation across different fields like machine learning and healthcare where many private data are involved. In a typical scenario where secure computation is needed, a client sends some data which could be private to a remote machine where it is processed and the results are sent back to the user. In this process, the client data should need to be protected and should be oblivious to the remote server. For example, a client can send a CT scan for diagnostic purposes to a hospital machine, where a series of trained models of various patients could be used for machine learning prediction. In this example, the client would want to avoid disclosing his personal data while still getting some useful diagnosis. As these data and services become more advanced and readily available, there is an increasing need to keep private data secure even when stored in the cloud and still used for research and testing. Genetic information, for example, has numerous special distinguishing features and it can violate personal privacy via genetic disclosure or genetic discrimination. Due to these potential privacy issues, there is a great need for a protocol for the secure outsourcing of private data analysis in a cloud environment.

To enable these applications, secure computation technologies such as multiparty computation and homomorphic encryption have been proposed and studied but have not had widespread deployment because they present practical challenges. Multiparty computation (MPC) is a secure computation protocol that allows for computation of a function by a set of parties who possess private inputs that are not revealed to other parties. The most common form of MPC is two-party computation (2PC) which is usually used for secure outsourcing of private computations of a client to an untrusted cloud machine. Yao's Garbled Circuit (GC) and Goldreich-Micali-Wigderson (GMW) are representative 2PC protocols. Homomorphic encryption (HE), on the other hand, supports operations on encrypted data thus making it possible for data to remain confidential while it is processed in untrusted environments [1, 2]. This property allows for the protection of private data especially in cloud services. Furthermore, the most mainstream solution are trusted execution environments (TEE) like Intel SGX which provide a bubble environment where data is protected and can be used for secure computation. All of these have different security assumptions and thus offer different security guarantees and corresponding implementation overheads.

As these solutions become more common, datacenter would have to be equipped with the necessary hardware and software to support them. This puts additional burden on datacenters which already expends a lot of resources to host emerging applications such as machine learning. For example, datacenter power consumption has been increasing significantly in recent years[3]. Aside from enabling the deployment of these solutions, it is also important to ensure that the datacenter implementing these solutions will have reasonable overheads. This presents an opportunity for architects for proposing solutions that will make these applications practical in terms of performance and datacenter overheads as evidenced by many recent advancements in making secure computation practical through implementing accelerators [4, 5, 6].

Meanwhile, as Moore's Law and traditional device scaling ends, the push for continued systems performance scaling becomes even more challenging. While architects have recently exploited chip specialization to compensate for limited device scaling [7], this too has its own limitations[8], which encourages exploring device technologies beyond traditional Silicon-based CMOS, badly needed in to address datacenter overheads. Steep-slope devices such as negative capacitance FET and superconducting electronics (SCE) are promising "Post-Moore" options that promises lower energy and potentially higher performance compared to traditional silicon-based CMOS. Superconducting devices have been well-studied and there have been many proposed variations such as rapid single-flux-quantum (RSFQ) [9], energy-efficient RSFQ (ERSFQ) [10], and more recently, adiabatic quantum-flux parametron (AQFP)[11]. While significant progress have been made in advancing the material, device and circuit properties, there has been little progress in understanding the architectural and system-level implications of these technologies, especially in the the context of datacenter hardware and applications, where they are badly needed.

To help address these issues, in this dissertation, we study new opportunities and tradeoffs and propose solutions for architectural support of emerging datacenters workloads such as privacy-preserving computation. In addition, we leverage high-level modeling techniques for early and rapid evaluation of Post-Moore technologies for potential datacenter hardware accelerator and processor designs.

In particular, in the following three chapters, we present different solutions that can help tackle the challenges of the emerging datacenter privacy and security-focused workloads using both architectural specialization and technology advancement using Post-Moore devices.

- In Chapter 3, we study the trade-off of physical security and performance improvement by improved locality in the context of multiparty computation in datacenters. We propose an asymmetric approach to multi-party architecture with the co-location of a small physically-hardened compute element (under the control of one party) with a much larger and robust server-class system (under the control of the other). We call our proposed devices "Hardware Embassies", a new class of devices that enable more efficient MPC by providing untrusted server co-located tamper-proof trusted hardware.

- In Chapter 4, we study the trade-off of simplified computation but at the expense of larger data to be processed in the context of homomorphic encryption-based search in datacenters. We propose a near-data processing (NDP) architecture using 3D-stacked DRAM to support privacy-preserving biomarker search which enables reduction of data movement and acceleration of basic additive homomorphic operation.

- Lastly, in Chapter 5, we study the benefits and drawbacks of Post-Moore technologies in the context of hardware (processors and accelerators) starting to be introduced in datacenters for these applications. We use rapid, early-state statistical and analytical models to explore the performance and power benefits of steep-slope devices and superconducting technologies. Our work serves as early guidance of the the limitations and potential of these Post-Moore technologies through performance and energy efficiency design space exploration.

Chapter 2

Background

In this chapter, we provide background information on secure computation, in particular, on Multiparty Computation (MPC), Homomorphic Encryption (HE), and Trusted Execution Environment (TEE). We discuss recent advancements as well as compare the characteristics, emphasizing the the pros and cons of each. We also describe privacy-preserving applications that make use of these techniques. Finally, we describe emerging technologies such as steep-slope devices an superconducting electronics and their potential applications.

2.1 Secure Computation

2.1.1 Cryptographic Primitives and Protocols

Multiparty Computation (MPC)

Multiparty computation allows for computation of a certain function by a set of parties who possess private inputs without having to reveal these inputs to other parties. In this paper, we are interested in two types of MPC protocols, Yao's Garbled Circuit and Goldreich-Micali-Wigderson. Before providing a brief overview of the two protocols, we first discuss their underlying primitives. The most common implementation of MPC is a two-party computation (2PC) which usually involves a client and a server. Two adversary models are typically used for MPC: *semi-honest* (honest-but-curious) wherein all parties follow the protocol but the attacker may try learn more than what is allowed, and *malicious* wherein the attacker is not restricted to follow the protocol and so can actively attempt to compromise security.

Oblivious Transfer

Oblivious transfer (OT) is a fundamental cryptographic primitive heavily used in secure two-party computation. 1-*out-of*-2 *oblivious transfer*, is a protocol in which a sender inputs two messages (x_0, x_1), each of m-bit size, and a receiver inputs a choice bit $c \in \{0, 1\}$, such that the receiver obtains the message x_c without knowing x_{1-c} and the sender does not learn any information about the choice of c requested by the receiver. There is a variant of OT called a *random oblivious transfer* that reduces communication overhead. In a random OT, the sender inputs no messages and obtains a random message pair from the protocol itself. The receiver still inputs its selection bit to choose one of the random messages. 1-*out-of*-2 OT can be generalized to 1-*out-of*-N OT (N $\geq$ 3) where the sender inputs N messages of m bits instead of two messages. Again, in 1-*out-of*-N OT, the receiver inputs a choice bit string $s \in \left[1, n\right]$ to obliviously obtain the message x_s. Note that regular OT still requires some public key cryptography, which is known to be costly.

OT Extensions

[12] shows that it is possible for both parties to compute n OTs using only symmetric cryptography after computing κ public-key base OTs, where $\kappa \ll n$ and κ is the the security parameter. This technique is called OT extension [12] and is effective because symmetric key operations are much faster than public key operations. Given that modern

Intel processors are equipped with AES-NI extensions to enable hardware acceleration in place, this is an obvious improvement in computation compared to calling regular OT n times. The sender and receiver of an OT have asymmetric computations. An OT requires the sender to compute slightly more intensive cryptographic functions of AES256 or SHA256. A more optimized OT extension, described in [13], utilizes random OT to reduce communication overhead. As a result, the implementation is able to achieve computing over one million OTs per second [13].

OT Overhead

For simplicity, we list communication cost that includes both data sent and received. With OT extension, the communication is $2m + \kappa$ bits for a 1-*out-of*-2 OT and κ bits for a 1-*out-of*-2 random OT [14, 13]. The communication is $Nm + 2\kappa$ bits for a 1-*out-of-N* OT and 2κ bits for a 1-*out-of-N* random OT [14]. A 1-*out-of*-2 OT requires the receiver to compute two pseudo random generator (PRG) and one correlation robust function (CRF) and requires the sender to compute one PRG (AES128) and two CRF (AES128) evaluations. A 1-*out-of-N* OT requires the receiver to compute four PRG and one CRF and requires the sender to compute two PRG and N CRF evaluations [15]. CRF here is usually instantiated with either SHA256 or AES256. Because a CRF function is a lot more costly than a simple PRG using AES128, the sender role in OT has more computation overhead. receives a value of his choice from among several values sent by the sender, while learning nothing about the other values. The sender does not learn anything from the protocol, and in particular he does not learn which of the values he sent was received by the chooser.

Secret Sharing

In secret sharing, a value is shared between two parties such that the addition of two secrets yields the actual value. In order to additively share a secret x, a random value r is generated as party P_1's share, denoted as $[x]_1$, and $x - r$ is used as party P_2's share, denoted as $[x]_2$. To reconstruct a secret, one party needs to send its share to the other party so that the other party can add the two shares. Secret sharing is typically in the binary field. Because XOR is reversible in the binary field, one can simply use the random bit r and $x \oplus r$ as the shares of the two parties respectively, since $(x \oplus r) \oplus r = x$.

Yao's Garbled Circuit (GC)

Yao's garbled circuit [16] is a commonly-used two-party secure computation protocol. A function f is represented as a Boolean circuit composed on two-input gates like AND and XOR. P_1 and P_2's inputs are represented as input wires of the circuit. The goal of the protocol is to compute the circuit such that only the circuit output wires are revealed and values obtained in all other wires are not. To execute the protocol, the two parties take up the role of a garbler (P_1) and evaluator (P_2). In the garbling phase, P_1 creates a garbled table for each gate by first assigning two random labels to each wire in the circuit. The size of the label is determined by the security parameter which is typically set to 128. Each encrypted row in the table is obtained by double encryption of corresponding input labels. This means to decrypt the output of each gate (table), the two keys corresponding to the input labels are needed. P_1 sends these gabled tables as well as her input labels to P_2 for evaluation. In order to execute the evaluation phase, P_2 needs to obtain his corresponding input labels from P_1. This would require 1-out-of-2 OT described earlier since (1) P_2 cannot send his actual input to P_1 and (2) P_2 cannot send both input labels (for both 1 and 0) to P_1 as this would allow leakage of information. Once P_2 gets his input labels, he can start the evaluation. This involves decrypting the correct output key for each gate sequentially using the two input labels from both parties. This is done until the output wires are reached. The output mapping from P_1 is used to translate the output wire results to proper plaintext outputs.

Garbled Circuit Optimizations.

We summarize the most important garbled circuit optimizations below:

- *Free-XOR [17]*: Enabled free computation of XOR gates.
- *Row-Reduction [18]*: Reduction of garbled table rows from 4 to 3.
- *Garbling with Fixed-Key Block Cipher [19]*: Encryption of garbled tables using AES which can be efficiently executed in processors with AES hardware support (AES-NI).
- *Half-Gates [20]*: Further reduction of number of rows in AND gates from 3 to 2.
- *Sequential Garbled Circuit [21]*: Reduction of circuit memory footprint by using much smaller sequential circuits (executed in multiply clock cycles) instead of bigger combinational circuits.

Goldreich-Micali-Wigderson (GMW)

Similar to GC, the GMW protocol [22] allows two parties to securely evaluate any function represented as a Boolean circuit with two-input gates without leaking each party's private inputs. GMW is also based on the secret sharing scheme in binary field. All input values of the Boolean circuit are secret shared between the parties so that each party can evaluate the circuit using its share of each wire. Computing an XOR gate can be done locally without any communication. Assume two parties have a share of $x = [x]_1 \oplus [x]_2$ and $y = [y]_1 \oplus [y]_2$. To secret share $z = x \oplus y$, it is sufficient for P_1 to compute $[z]_1 = [x]_1 \oplus [y]_1$ and for P_2 to compute $[z]_2 = [x]_2 \oplus [y]_2$, as $z = [z]_1 \oplus [z]_2$. Meanwhile, evaluating an AND gate requires each party to use a pre-computed multiplication triple generated for each AND gate. GMW is divided into two phases, a setup phase and an online phase. The setup phase computes each multiplication triple using 2 1-*out-of*-2 OTs so that the evaluation of AND gates in the online phase can be performed efficiently. Note that multiplication triples can be generated before the actual function is known but it still requires communication between both parties. As a result, both parties need to stay online for the entire setup phase in the traditional setting. Because XOR gates are localized, the only communication takes place at each AND gate. To reduce communication rounds, the two parties can compute AND gates in the same circuit level in parallel.

Because XOR gates are localized, the round complexity of GMW only depends on the depth of AND gates in the circuit. The downside of this dependency is that the performance of GMW can be sensitive to the network latency. Using a circuit with lower depth and greater size can be more efficient for GMW.

LUT-based protocols represent the evaluation function as a Boolean circuit with multi-input gates. The circuit is synthesized to a network of lookup tables (LUT) as oppposed to just two-input gates, which makes the circuit more compact and enables the evaluation of more intensive functions. The state-of-the-art LUT-based protocol [23] reduces the communication by a factor of 4x compared to GC and reduces the round complexity by a factor of 4x compared to GMW, at the cost of slightly increased computation. [23] proposes two LUT-based protocols, SP-LUT and OP-LUT, which are optimized for setup phase and online phase respectively. Both of the protocols are based on binary secret sharing, meaning that they are fully compatible with GMW and mixed protocol schemes can be further explored.

Homomorphic Encryption (HE)

Homomorphic encryption (HE) is a cryptographic tool that allows performing computation even when the data is encrypted. With a homomorphic encryption scheme, performing the function f on the encrypted data will result to the value as with performing encryption first, evaluate the function f on the encrypted data (ciphertext) and

Optimization	Size per gate		Calls to H per gate			
			Alice		Bob	
	XOR	AND	XOR	AND	XOR	AND
Classical	4	4	4	4	4	4
Point-and-permute	4	4	4	4	1	1
GRR3	3	3	4	4	1	1
FreeXOR	0	4	0	4	0	1
GRR2	2	2	4	4	1	1
FleXOR	$\{0, 1, 2\}$	2	$\{0, 2, 4\}$	4	$\{0, 1, 2\}$	1
Half Gates	0	2	0	4	0	2

Figure 2.1: Garbled Circuit Optimizations [20]

decrypting the ciphertext. Homomorphic encryption systems can be divided into additively homomorphic such as Pallier or multiplicatively homomorphic such as RSA. Gentry provided the first fully homomorphic encryption scheme [24] but, in practice, partially homomorphic and leveled homomorphic encryption schemes are used because of computational efficiency. Compared to multiparty computation, homomorphic encryption has the advantage of being non-interactive since it does require the client to be online to perform the computation which reduces network overhead as well as not needing additional servers. Currently, because of its significant computation overhead, it is usually not used in isolation but is usually paired with MPC in hybrid protocol solutions [21].

Trusted Execution Environment (TEE)

As an alternative to multiparty computation and homomorphic encryption, trusted execution environments (TEE) built into recent processors can also be used for secure computation. TEEs such as Intel SGX and ARM TrustZone have secure enclaves which provides functions such as isolated execution environment for secure computation and remote attestation of code running within enclave to ensure integrity. Although they have lowest performance overhead compared to other secure computation methods and currently deployed in real world products, recent attacks on SGX (Foreshadow [25]) and TrustZone [26] has put into question their security.

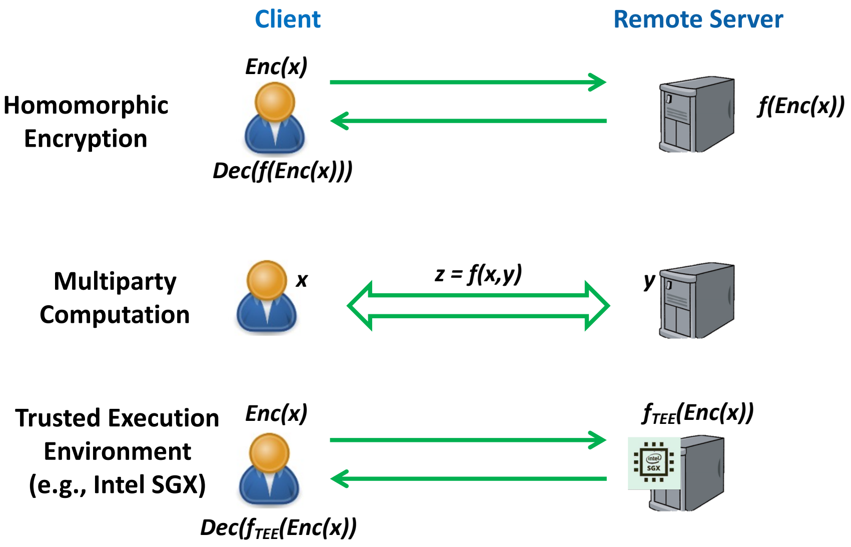


Figure 2.2: Comparison of MPC, HE, and TEE

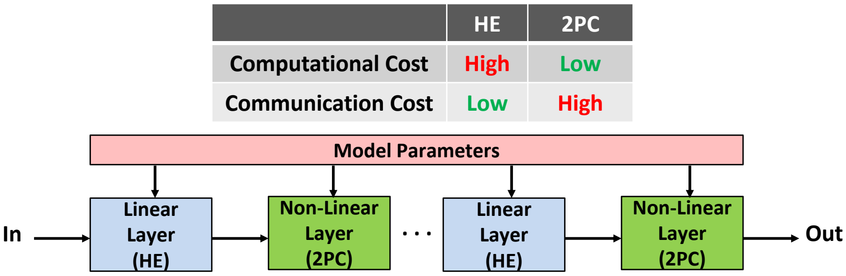
2.1.2 Privacy-Preserving Applications

Secure Neural Network Inference

Privacy-preserving neural network (NN) inference is one the most active research applications of secure computation [27, 28, 29, 30, 31]. There has been a steady increase of interest in NN in recent years which has allowed significant advances in computer vision, natural language processing, and other fields. However, most NN applications currently in use have so far focused on efficiency and have put privacy in the backseat. Data transmitted from the client which are used for prediction/inference in a cloud provider could be of sensitive holding private information such as financial and medical information. Outsourcing of prediction carries the risk of client sensitive information being leaked out. Thus, there is a great need for privacy-preserving NN inference enabled by secure computation.

Secure DNA Matching using Private Set Intersection

Personalized DNA analysis has become increasingly popular since the emergence of



	HE	2PC
Computational Cost	High	Low
Communication Cost	Low	High

Figure 2.3: Secure NN Inference Mixed-Protocol Flow[30]

23andMe, a company that offers DNA testing for ancestry. DNA testing not only helps customers connect with their ancestral relatives, but it also helps patients identify their susceptibility to certain diseases. However, privacy policies of DNA services may allow third parties to access sensitive genetic information from customers. Hence, there is a great need for secure DNA analysis such as genome matching through private set intersection (PSI). In the PSI application, two parties want to learn the intersection of their sets without revealing any private element not in the intersection. Because PSI needs to be applied to large sets of data, performance is critical. Since traditional secure implementations of PSI scale poorly with large data sets, many current implementations conform to an insecure protocol using a one-way cryptographic hash function. Moreover, many recent works have made PSI more practical by focusing on reducing communication cost [15, 32, 33].

Privacy-Preserving Biomarker Search using HE

Biomarker search is one of the key emerging applications in bioinformatics domain [34], as it allows for detection of possible diseases. A specific set of biomarkers are queried from a server that houses a database of these biomarkers. The presence or absence of a specific biomarker or a set of biomarkers indicates a probability of genetic diseases and thus helps medical practitioners to make informed decisions. In dealing with this type of application, however, data is stored in the database and the queries must be encrypted in order to protect privacy.

The biomarkers are stored in Variant Call Format (VCF). These VCF files contain information on biomarkers (genotype information) such as chromosome number and the position of the genome. Furthermore, it contains information for each position such as reference and alternate sequences.

A typical processing flow for HE-based biomarker search is shown in Figure 4.2. The figure shows two general phases: a preprocessing phase and the query phase. In the preprocessing phase, each entry in the VCF file is first encoded and hashed before performing the actual homomorphic encryption using a generated key. This is to reduce the size of the encrypted entries since the size of the unencrypted entries will affect the size of the data after encryption. In the query phase, the client similarly needs to preprocess the query before it is sent to the cloud service for the exact search operation. An encrypted result of the search is sent back to client where it can be decrypted using the secret key. In this work, we focus on the homomorphic evaluation stage of the search which takes up the majority of the execution time, especially for large number of queries.

2.2 Emerging Post-Moore Technologies

2.2.1 Steep-Slope Device Electronics

Steep-slope devices are a class of technologies that have significantly improved power-efficiency compared to traditional silicon-based CMOS technology. What these devices try to overcome is the is the fundamental limit of subthreshold-switching of CMOS-based devices (60mV/dec). In this section, we give an overview of one of the representative emerging devices, Negative capacitance FET (NCFET).

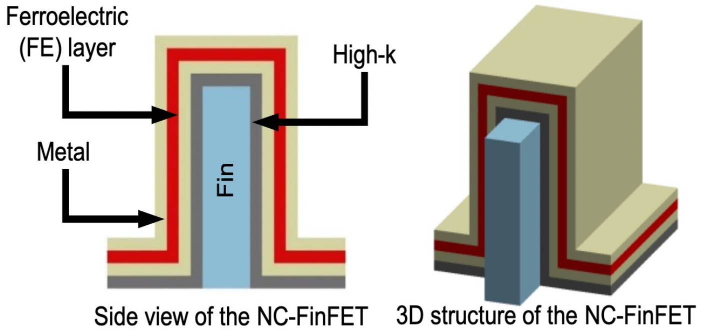


Figure 2.4: Negative Capacitance FET Structure [35]

Negative Capacitance FET (NCFET)

NCFET has an integrated ferroelectric layer within the transistor gate. Typically, the already used high-k dielectric HFO2 is doped by Zr to obtain the ferroelectric layer. As such, the device itself has not significant difference in terms of area compared to CMOS FinFET. A typical NCFET structure is show in Figure 2.4. This effect of this ferrorelectric layer is what is known as negative capacitance, which results in charge redistribution which ultimately lowers the required potential to switch the transistor compared to non-NC oxides. Thus, NCFET allows operation at a higher frequency at the same voltage or operate at the same frequency at lower voltage compared to CMOS FinFET [35]. One disadvantage of NCFET is that it increases the gate capacitance as a result of the negative capacitance effect. This results in higher dynamic power at the same gate voltage.

2.2.2 Superconducting Electronics

Superconducting electronics use materials such that at least some parts of which are in superconducting state. Since superconducting electronics require to maintain at certain temperature due to their unique physics characteristics, the common temperatures for superconducting devices are the boiling point of liquid nitrogen, the boiling point of liquid helium, and the superfluid helium-4 temperature, which is below 2.17K. Although this is a crucial setting for current computers, their performance and energy-efficiency are promising for the future post-Moore research in computer architecture. For example, 8-bit AQFP adder reported a 24 k_bT energy dissipation per junction [36] .

Superconducting technology is based on the Josephson Junction (JJ), a primitive switching device. A JJ is composed of an insulating barrier that is sandwiched between two superconducting layers. In its superconducting state, despite no voltage applied across the junction, tunnelling current can pass through the junction. Once a certain current limit (I_c, critical current) is flowing through the junction, it switches to its resistive state.

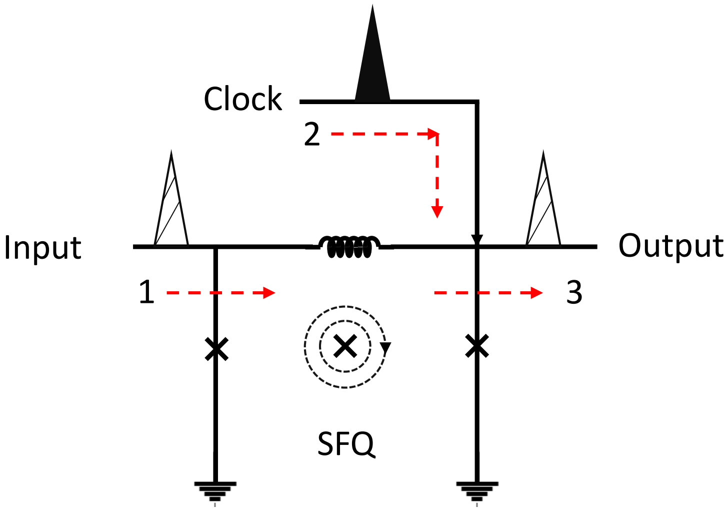


Figure 2.5: SFQ DFF

RSFQ and AQFP are two leading candidate superconducting technologies. The clock frequency for RSFQ can reach 50 GHz for 8-bit processors[37]. Although RSFQ can operate at high clock frequencies, it still has a significant leakage from its bias resistors used to supply DC currents. This drastically increases the static power dissipated between 10 and 100 times the dynamic power. An energy-efficient RSFQ (ERSFQ) was proposed to address this. In particular, ERSFQ differs from RSFQ by adding a large bias inductor as well as a clocked feeding JTL to limit its supply power to perform current-limiting Josephson junction to distribute bias power. Thus, this design decreased the energy consumption with extra area. The power distribution for energy-efficient operation adds about 30% to ERSFQ overhead [38].

AQFP makes use of AC bias for its clock and power supply, unlike other superconducting technologies like RSFQ which are based on DC bias, allowing it to avoid DC power overhead and essentially consume less power [39]. The most basic logic unit in AQFP is a buffer. Excitation fluxes are generated in the JJ loops as current I_x is applied. An SFQ (Single Flux Quantum) is then stored in either left or right loop depending on the input current I_{in}. Note that, compared to RSFQ, which use JJ switching to move SFQ, information in AQFP is encoded by the location of the SFQ, which determines whether it represents logical '1' or '0'. Inverter and constant cells can be generated from this buffer cell. This set of 3 cells can then be used to build logic gates such as MAJ (majority), NOR, and AND. As a result of having the same AC signal as a power source and clock, a clocking scheme is needed to synchronize the outputs of all gates in the same clock phase. Typically, individual AQFP logic gates are connected to an AC clock signal and each one will occupy a clock phase.

Chapter 3

Establishing Cooperative Computation with Hardware Embassies

While recent cryptographic techniques enable cooperative multi-party client-server computations under mutual distrust, they also introduce an efficiency tradeoff. Hosting all of the computation from the different parties involved on one set of servers requires everyone to agree on which servers are trustworthy. On the other hand, keeping the computations truly distributed introduces significant delays because of the inherently latency-sensitive nature of the protocols involved. In this work, we explore the architectural impact of a possible middle path to this problem: resource-poor but physically secure devices interacting with significant (but not mutually trusted) compute and storage resources. The idea is that a small and well-protected "Embassy" can serve as a plot of sovereign soil in an otherwise untrusted environment. Building on techniques from multiparty computation (MPC) we show how such an architecture, even when extremely limited in size, can leverage local network capabilities and asymmetries in cryptographic operations to perform more efficient interactive secure computations. Even with a client-side device 5× slower, we show that common MPC applications can still be accelerated by 3× on average. Moreover, we explore the potential for architectural changes to further support multi-party evaluation through the addition of dedicated evaluator hardware further improving performance 1.52×.

3.1 Introduction

Through advanced cryptographic techniques, it is now possible to perform shared computations without ever fully sharing the data. For example, a class of cryptographic techniques referred to as multiparty computation (MPC) establishes secure computation *protocols* between multiple non-colluding parties that allow for functions to be iteratively computed on private inputs without revealing anything beyond the result to either party. As long as we trust those parties do not share out-of-band information with one another, these techniques allow for a mutual computation to be performed (for example a query to a database) without either side learning what the other is doing (such as keeping the query secret from the database and visa versa). Unfortunately, these protocols usually require both parties to be *active participants* in the computation to some degree. Because the computations are typically arranged as long and unbroken chains of cryptographic operations, involving multiple parties typically means a lot of *waiting around* for the other side to finish up their work and pass it back to you.

One way to deal with this is to host multiple parties on a trusted third-party platform. Co-locating the computation minimizes the time wasted transmitting parts of the computation back and forth between all parties. Of course, if you had a fully trusted third party they could just do the computation for all parties involved – no need for cryptography! However, when we “trust a server”, we are trusting not only the computational and storage resources it hosts but also the **physical** and **legal** environments under which it operates. These aspects are hard to attest to remotely and only compound when multiple nation-states are involved. In reality, these cryptographic approaches are typically the most useful when the data involved is sensitive enough that we prefer to trust no one with all of the data. So, what can we do?

That introduces the new problem of where to find computational resources to host multiple parties that both parties will trust. The last decade has seen significant advances in making trusting third-party remote hardware a more reasonable choice. For example, Flickr [40] introduced a clever scheme for remote attestation built on the Trusted Platform Module (TPM) architecture which allowed the loaded system binary to be non-bypassably fingerprinted. More current approaches build on top of the capabilities of trusted execution environments (TEE) such as Intel SGX [41] to create similar "bubbles" of trust. While these and other approaches provide significant protections, the threat models one can address with ISA-level changes alone are constrained and the limits of sharing resources with an untrusted host opens up many potential side-channel attacks.

The question we attempt to answer in this paper is if and when it is *possible* to use a small island of physical security located in an otherwise very untrusted environment, to enable a broader set of physically secure computation. Moreover, we explore new architectures and machine organizations that enable such an approach to operate with higher efficiency and better performance as compared to remote computation.

Specifically, we propose an asymmetric approach to multi-party architecture with the co-location of a small physically-hardened compute element (under the control of one party) with a much larger and robust server-class system (under the control of the other). The hardened device can be physically smaller with fewer compute resources. Due in part to its small size, the small compute element can be hardened against even incredibly advanced attacks to a high degree. The small device can even be physically shipped between the guest, host, and back again as needed for initialization and decommissioning. At a high level, one can think of this idea as setting up an "*Embassy*" that serves as an island of sovereign soil in a foreign land. Just like traditional embassies, this arrangement allows for higher bandwidth and lower latency interaction facilitating joint activities even under mutual distrust. The code that lives on the device can serve to orchestrate and even *participate in* trustworthy computations in the server on behalf of the guest. Physically shipping the device adds significant setup overhead. However, there are many privacy-focused applications where this one-time cost is tolerable. For instance, in hospitals or research centers, new and sensitive data are being generated constantly. The more this device is used, the more amortized the setup cost becomes.

As a first demonstration of the concept, we identify a class of cryptographic computing approaches that are *inherently asymmetric* in their needs. Building on techniques from homomorphic encryption and multiparty computation, we show how our proposed system can leverage the high bandwidth and low latency network fabric available locally to perform more efficient interactive secure computations, even when the computational abilities of these physically-smaller devices are severely limited. Specifically, we examine two important privacy-preserving applications, secure neural network inference based on Yao's Garbled Circuit (GC) [16] and private DNA matching based on Goldreich-Micali-Wigderson (GMW) [22]. In these scenarios, the Embassy (our proposed device) acts as a trusted (non-colluding) proxy for the client to perform Multi-Party Computation (MPC) with a co-located untrusted server. We show that the improvements in connectivity possible from using only systems connected by local networks more than compensate for the smaller compute resources available to this new class of device, *and* that with some simple architectural changes this gap can be extended even further. We summarize our contributions as follows:

- We propose “Hardware Embassies”, a new class of devices that enable more efficient MPC by providing untrusted server co-located tamper-proof trusted hardware.

- We show how important cryptographic methods can be mapped to Hardware Embassies and, for the first time, quantitatively explore the ways in which we can take advantage of the network performance and asymmetric compute requirements of these protocols.

- We quantitatively evaluate two different important applications: secure neural network inference using a hybrid protocol (GC + HE) and private set intersection which can be used for private DNA matching using GMW protocol.

- Building on our experience with the above, we propose and evaluate a microarchitecture specialized in the cryptographic operations at the heart of common MPC computations.

- We show experimentally, through a mix of in-datacenter network experimentation, detailed simulation, and Verilog design, that the resulting system realizes a 4.56× improvement over more distributed computation.

We start with a discussion of MPC and its objectives in Section 4.2. We discuss the basic overview of secure computation protocols and the motivation for our work. In Section 3.3 we present details of our solution. We evaluate our proposed solution in Section 3.6 and provide a discussion of related literature in Section 3.7.

3.2 Supporting MPC

Multi-Party Computation (MPC) is a class of cryptographic techniques that allow for the evaluation of functions without any of the participating parties learning about the inputs used in the computation [42]. The most advanced techniques support any computation expressible as a Boolean circuit, everything from neural network evaluations to bioinformatics applications, without sharing the underlying data.

A common form of MPC in practice is two-party computation (2PC) [43], which can be used as a way to securely outsource private computations to untrusted cloud machines. Yao's Garbled Circuit (GC) and Goldreich-Micali-Wigderson (GMW) are examples of 2PC protocols which have been used for applications such as privacy-preserving machine learning [44], secure genomic computations [45], and secure data search [23]. Recent algorithmic improvements to 2PC protocols, especially the transition from public-key cryptography to symmetric cryptography, have reduced the computational overhead by more than an order of magnitude but communication bottlenecks are much harder to overcome. For a single inference operation in a simple MNIST-based neural network, a strict GC approach would require a network transfer volume of 791MB [46]. While we will discuss some algorithmic ways others have found to help mitigate this problem, it remains a serious issue.

Also, the literature on MPC is largely dominated by work that optimistically assumes direct high speed and low latency connection between communicating parties. The high communication cost of GC becomes even more burdensome for the common case where a client and a server are located in different regions and therefore are using a WAN connection. There are different possible LAN and WAN assumptions one might make, but typically settings of WAN have 430× longer latency and 113× smaller bandwidth than the reported LAN configuration for AWS [28].

We propose the use of a low-resource device under the direct control of an entity cooperating with the co-located server that takes advantage of the high-speed LAN performance. This is made possible by *physically co-locating this device* with the cooperating agents while taking advantage of the inherent asymmetry of client and server computa-

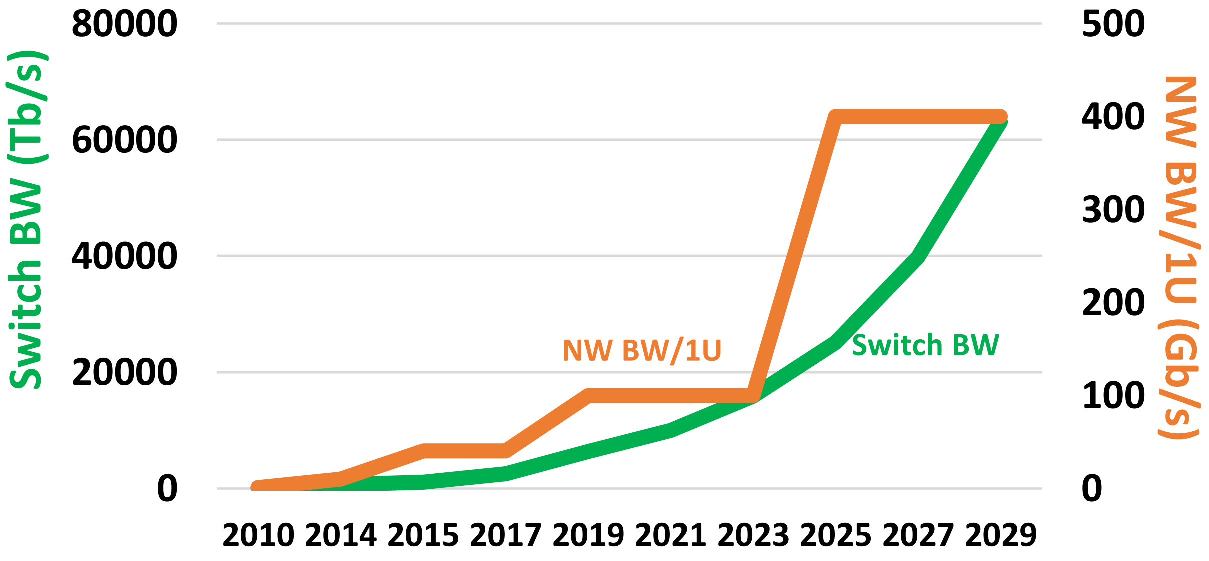


Figure 3.1: (a) Speed of 2PC Improvement [47]. Due to several optimizations in two party computation in recent years (see Section 2.1.1), the network communication has become the key bottleneck. (b) Datacenter network trends [48]. Network bandwidth available to server units and total switch bandwidth of datacenters is expected to increase. It is yet to be leveraged in secure computation applications.

tional requirements for most common cryptographic techniques. For example, as shown in Table 3.1, the evaluation phase in GC (typically performed at the client-side) has 2× smaller compute requirements [20] than the garbling phase (typically performed on the server). This difference in compute load between the client and server becomes more asymmetric for the hybrid protocols we consider in this work.

Unfortunately, co-location inherently creates a trade off between performance and security as one party now has physical access to both sides of the computation. In situations with mutual trust between all parties, this does not pose a security challenge, however, under these assumptions it is often unnecessary to utilize a co-located device. When the embassy device is under the physical control of an untrusted entity then that entity could potentially break the non-collusion assumptions that MPC and other cryptographic protocols rely on. It becomes necessary to ensure the embassy is secure against physical attacks.

While physical security is not the focus of this work, NIST provides a standard for the security of cryptographic hardware in untrusted environments called FIPS 140. The latest versions, 140-2 [49] and 140-3 [50], categorize hardware into four categories. For FIPS 140-3 levels 1 and 2 have no physical security requirements, and so such devices would be unfit for an embassy device. Levels 3 and 4 require strong enclosures with tamper detection that causes either an automatic zeroisation or a module shutdown. While both level 3 and 4 devices are sufficient to implement Embassy, these tamper detection techniques introduce overhead proportional to the original chip area. Also, there is rarely a single technique that is able to provide catch-all tamper detection [51]. For instance, silicon light sensors have been used to detect active optical attacks [52], but cannot detect other attacks. Due to this, most FIPS 140 level 3 or 4 devices tend to be small, such as USB drives, security cards, and hardware security modules. While these devices are too small to support the computation necessary for an Embassy, the techniques used can be expanded to cover a larger device.

Using a small computing device for the Embassy gives us the following advantages: (i) better defence against physical tampering because of a smaller attack surface; (ii) better protection if the server gets compromised, given the Embassy has a different hardware configuration and security guarantees (an attack on the server does not automatically compromise the Embassy); and (iii) this setup relaxes the need for the client to be online because it allows precomputations that can reduce ad-hoc runtime.

In this paper, we study two different applications to demonstrate the practicality of our solution. The first application is secure NN inference using a hybrid protocol (HE + GC) [30]. The second application is secure DNA matching using a private set intersection with the GMW protocol [23]. While we use these two specific MPC approaches to evaluate this approach, there is nothing application-specific about the architecture we propose.

For the two applications considered in this work, we follow the threat model of [30, 23].

In secure neural network inference, we assume that the network model is available as plain text in the server, similar to past work [30, 27, 29]. We assume that the cryptographic protocols which we make use of in this work are correct and that the adversary is computationally-bound, i.e. brute-force attacks are infeasible. We assume that the Embassy is resistant to physical tampering and that any attempt to pry open the device results in irretrievably corrupting the data in the device as per FIPS 140-3 level 3 and 4 devices.

3.3 Hardware Embassy Approach

The general protocol governing the use of an Embassy consists of three main phases.

I. Key Setup. Unique among other approaches, a client can begin with direct physical control of the device to be embedded in the co-located server. The client generates a random symmetric encryption key which is then stored in the Embassy. This key can be used to securely communicate back with the client from the co-located server.

II. Program Select and Compute. After the Embassy is installed in the co-located server, the client can send a request to the device consisting of an input and a program for the computation. This is done through a secure channel using the key that was generated by the client. To initiate the compute operation, the Embassy sends a request to an untrusted server in the system. For example, to perform GC, the untrusted server sends the garbled tables of the program to the Embassy and performs OT for input wires. Note that the garbled tables can be precomputed offline for certain programs. The Embassy can then evaluate the garbled tables and obtain the result of the computation.

III. Result Retrieval. As results are generated, the Embassy can send them to the client. Alternatively, the client can batch a request of computations and query the results stored in the Embassy. Results are sent back to the client using a secure channel

Properties	Garbled Circuit	GMW
XOR Gate	free	free
AND Gate		
- Setup computation	-	Client/Server: 6×AES
- Setup communication [bits]	-	C to S and C from S: 2×κ
- Online computation	C: 2×AES; S: 4×AES	negligible
- Online communication [bits]	C from S: 2×κ	C to S and C from S: 4
wire storage [bits]	κ	1

Table 3.1: Comparison between Garbled Circuit and GMW. The security level κ is usually fixed at 128 bits. XOR gates require no communications for both protocols and can be computed locally. For each AND gate, the garbled circuit computes more AES on the server-side (2×) while work is evenly split in GMW. In general, GMW requires more AES computations while GC consumes a greater memory footprint. Both protocols have the same communication overhead.

as the data leaves the co-located server.

3.4 Embassy Design

In designing an Embassy, we consider a spectrum of specifications with the highest performance being a server-class machine. Given the highly-advanced threat model we are considering, it would be advantageous to use a device that has small enough dimensions to be physically protected using the most aggressive tamper-resistant methods known [53]. At the lowest end that may be a simple USB-sized package similar to those used for edge neural network acceleration [54, 55]. However, we are interested in using a standalone device that does not need a host, for security and performance reasons. The closest commercial device on the market is Intel's compute stick which was first released in 2015 [56]. These devices have USB ports that can be used to connect to either a USB-based NIC or a switch that incorporates USB connectivity, however more complicated wire connectivity and better networking capabilities could be possible.

One of the main challenges in using such small devices as an Embassy is the lower

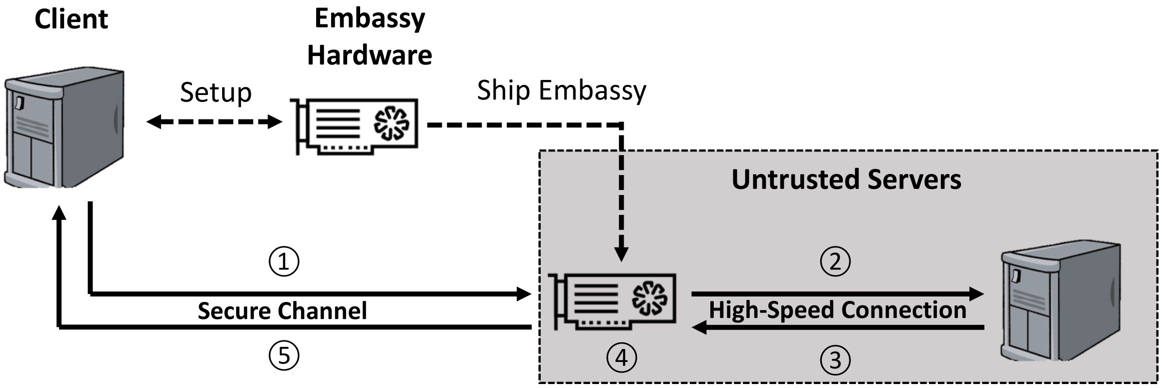


Figure 3.2: Protocol Flow. A trust setup phase is first performed with a client machine before being sent to the co-located server. The client machine sends inputs and receives results from the Embassy through a secure channel. The actual secure computation happens inside the co-located server between the Embassy and an untrusted server.

performance they provide compared to server-class machines. However, as we will show later, with some creativity this level of device can still provide sufficient compute for secure computation due to the fact that most of these protocols have an inherent compute asymmetry – most of the compute-intensive actions can be carried out on a powerful but untrusted server.

3.4.1 Co-locating with Untrusted Servers

Here we present one possible design for a co-located server that supports Embassy. We use the concept of a *disaggregated co-located server* that allows different computing devices or accelerators to be separated and individually addressed instead of relying on host machines [57]. A sample co-located server configuration with Embassy is shown in Figure 3.3. This design presents advantages in terms of cost, performance, and security. An Embassy without a host machine yields significant cost savings and lower maintenance costs. In terms of performance, it has fewer network and software layers to traverse since it does not communicate through a host machine. Using a compute-stick class device also

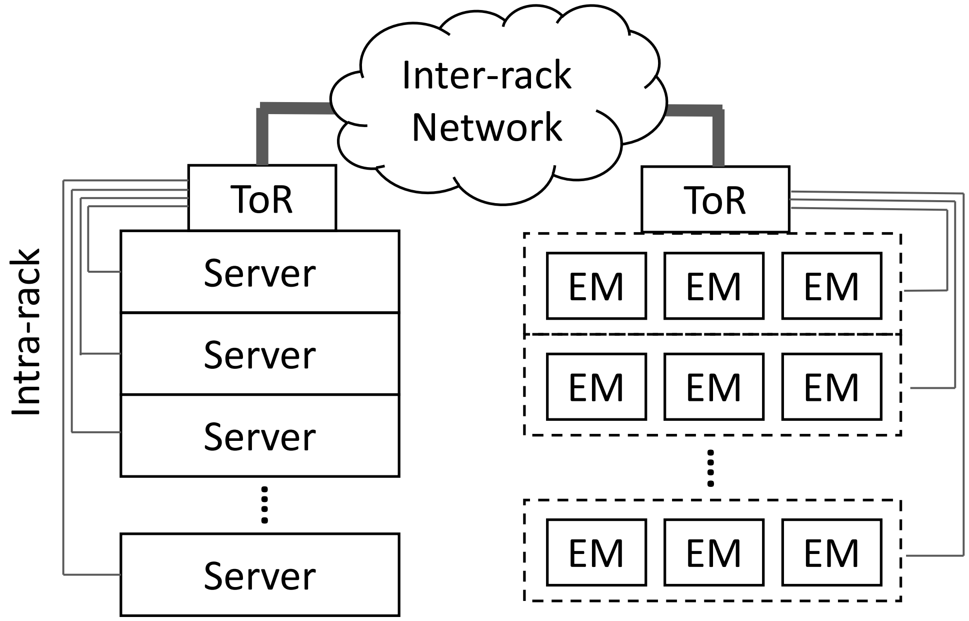


Figure 3.3: Model of Embassies with Co-located Servers. Embassies are host-less network-connected computing devices. Each one can connect to a server on other racks through Top of Rack (ToR) switches or through other Embassies.

ensures we are not over-provisioning for workloads that work on these devices. As for security, the potential attack surface is reduced since data does not need to pass through host machines, thus making side-channel attacks harder.

Dedicated and physically-separated machines for clients might be less cost-efficient than co-located servers. Nevertheless, recent attacks on virtualized environments [58, 59, 60] have made it clear that it is sometimes better to use separate machines if security is important since they can be more easily isolated in physically secure spaces. We also draw inspiration from the rise of baremetal servers which allow companies to have physically separate machines in the co-located server instead of using virtualized environments.

While we have shown that there are advantages in introducing third-party hardware such as Embassy into co-located servers, an understandable concern from server operators is if the device itself is malicious. While this has been an ongoing trend [61], here we discuss further potential safeguards to protect servers from malicious Embassies. One protection is to add a firewall using a switch exploiting software-defined networking to provide software-controlled protection of the broader co-located servers from potentially malicious traffic produced by errant Embassies. Another safeguard would be for the provider to release an open-source reference design for the Embassy with auditing by developers and the potential to perform attestation using Physical Unclonable Function (PUF) or Zero Knowledge Proof (ZKP) so that the provider can confirm that the Embassy can be trusted. The OpenTitan project shows a potential proof of concept in the related space of providing an open-source silicon root-of trust [62].

3.5 Applications

While Embassy can be used for a wide range of applications, in this paper we investigate two representative applications commonly outsourced to third-party co-located servers that highly demand privacy guarantees: neural network inference and DNA matching. In Chapter 2, we give an overview and motivation of these two applications and discuss related cryptographic primitives and protocols for those unfamiliar with the work. Below we describe in more detail how we can adapt these applications to leverage Embassy.

3.5.1 Embassies in Secure Computation

Embassy for Secure NN Inference: The protocol followed in this work is broadly similar to the hybrid protocol used in Gazelle [30], as shown in Figure 3.4. After securely receiving the input data from the client, Embassy encrypts the input data sent by the client (e.g., image) using packed additively HE (PAHE). The linear layers (convolution and fully-connected) are then processed using PAHE operations. Non-linear layers such as ReLU and MaxPool are performed using a garbled circuit. The ReLU circuit that is evaluated is shown in Figure 3.5, where s_x and s_y are shares from the server and c_x comes from the Embassy or client, and p is the prime parameter selected in PAHE. Conversion from PAHE to GC is done using secret sharing (adding a blinding random number). These steps are repeated in the series of linear/non-linear layers of the neural network until the final result (prediction) is obtained which is still in an encrypted form. This is sent back to the Embassy where it is decrypted and sent back to the client in a secure channel or stored for a later query by the client device.

Unlike in Gazelle, we assign the untrusted server as the garbler and the Embassy as the evaluator. In this way, we are taking advantage of the workload asymmetry that exists between the evaluator (less work) and garbler (more work). Note that compared to using pure GC, this hybrid protocol results in reduced online execution time and communication cost. This means that for cases where we are interested in similar runtimes, there is a larger margin for the performance degradation range of the Embassy.

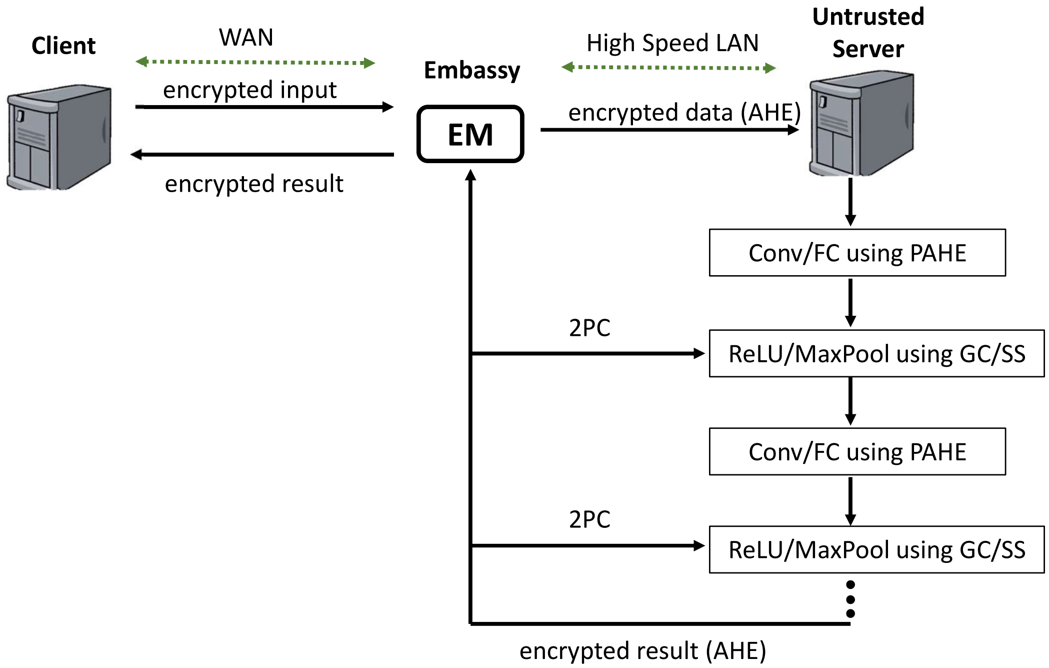


Figure 3.4: Hybrid secure neural network inference flow using Embassy. This flow is adapted from Gazelle [30] which combines Additive HE and Garbled Circuits to evaluate linear and non-linear parts of the neural network, respectively

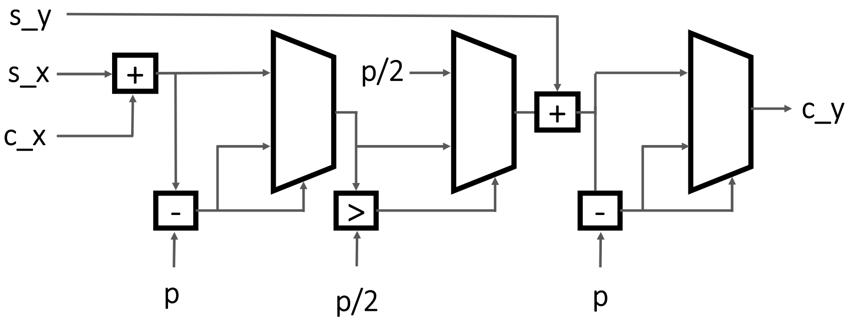


Figure 3.5: ReLU Gadget Unit to be evaluated in the GC phase of the Hybrid Secure Neural Network [30]

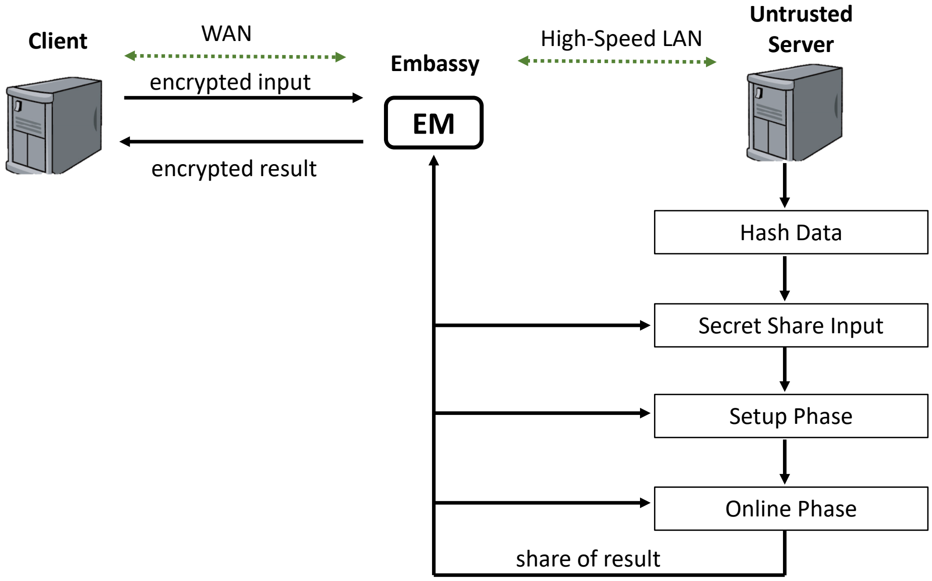


Figure 3.6: PSI-GMW Flow using Embassy. This is adpated from [15] that uses GMW to perform pairwise comparison for each bucket of the hash tables.

Embassy for Private Set Intersection: By improving an application using a generic circuit protocol alone (GMW), we demonstrate that similar results can be obtained in the most secure 2PC applications as GMW can be rapidly adapted to a different program by constructing a new corresponding circuit. We thus adapt the PSI pairwise-comparison-circuit using GMW[15] to Embassy. Dessouky et al. proposed a look-up table circuit protocol [23] that outperforms GMW in PSI, but since the protocol reduces the communication overhead at the cost of increased computation, it performs poorly in the LAN network and is thus not considered. For comparison, we evaluate a dedicated PSI protocol using Oblivious Transfer [32], one of the fastest PSI protocols in the literature, on the Embassy. For simplicity, we name the two protocols PSI-GMW and PSI-OT.

PSI-GMW computes the intersection between two sets by mapping elements from both parties into hash tables and evaluates a pairwise comparison circuit between each bucket of the hash tables, as shown in Figure 3.6. The complexity of PSI-GMW scales linearly with the product of the entry bit width and the set size [15]. The process begins with the client hashing its private data locally (e.g. genomic data in a VCF file) and sending that data to the Embassy in the co-located server. We adopt the same hashing technique in [15] that maps data to fewer bits, which reduces the one-time communication overhead over the WAN network and the storage requirement in the Embassy. The setup phase of GMW is dominated by AES operations in OT. To balance the computation workload, the server and the client switch roles in 1-out-of-2 OT after computing multiplication triples for half of the AND gates [63]. Because of the resource constraints of the Embassy, we remove this optimization and make Embassy always play the receiver in OT. Removing role switching also reduces computation intensity by reducing two base-OT computations to one. We still keep the two-thread implementation in [15]. Note that it is possible to improve performance by using more threads in the setup phase to further take advantage of the fast High-Speed LAN.

The Embassy enables the client to stay online only during the transfer of the input and output data. The multiplication triples can be generated as long as the size of the circuit is known. The actual program (e.g. PSI) does not have to be known in advance. Because the Embassy can always stay online, the Embassy can precompute a certain number of multiplication triples (say 2^{30}) with the server when it is idle. After the client makes a request to execute a program, it needs to query a multiplication triple for each AND gate in the circuit and uses them directly in the online phase. As a result,

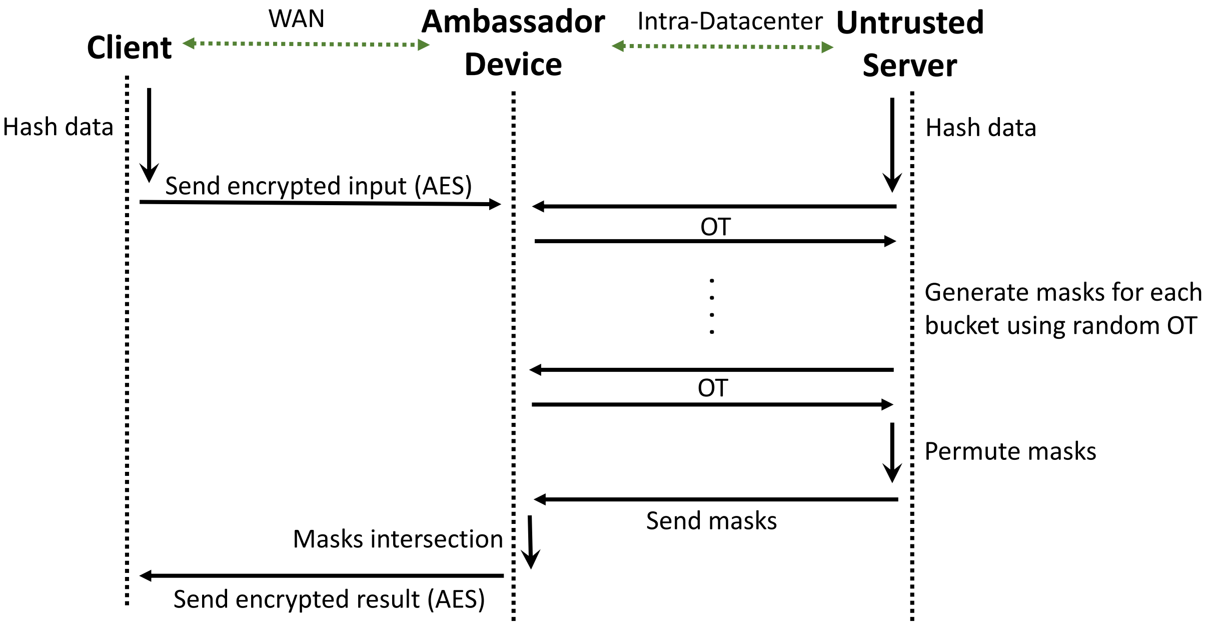


Figure 3.7: PSI-OT Flow using Ambassador. This is adapted from [32] that uses OT to generate masks and perform masks comparison for each bucket of the hash tables.

the ad-hoc runtime can be reduced by more than 99% for a set size of 100K. After the set intersection is computed, the intersection results will be stored in the Embassy in a bitmap format, which can be later queried by the remote client.

The PSI-OT flow, as shown in Figure 3.7, the same hashing process is still required. Instead of evaluating a circuit, both parties perform a random 1-out-of-N OT for each bucket of the hash tables. As a result, both parties obtain a randomly generated mask for all of their own table entries. Then the server sends a randomly permuted set of all of its masks to the Embassy. The Embassy finally computes the intersection by comparing the masks, and the results will be stored in the same bitmap format. The complexity of PSI-OT is independent of the entry bit width and scales linearly with the set size [32]. However, 1-out-of-N OT requires more base-OT computations, which can easily become compute-bound in a fast network.

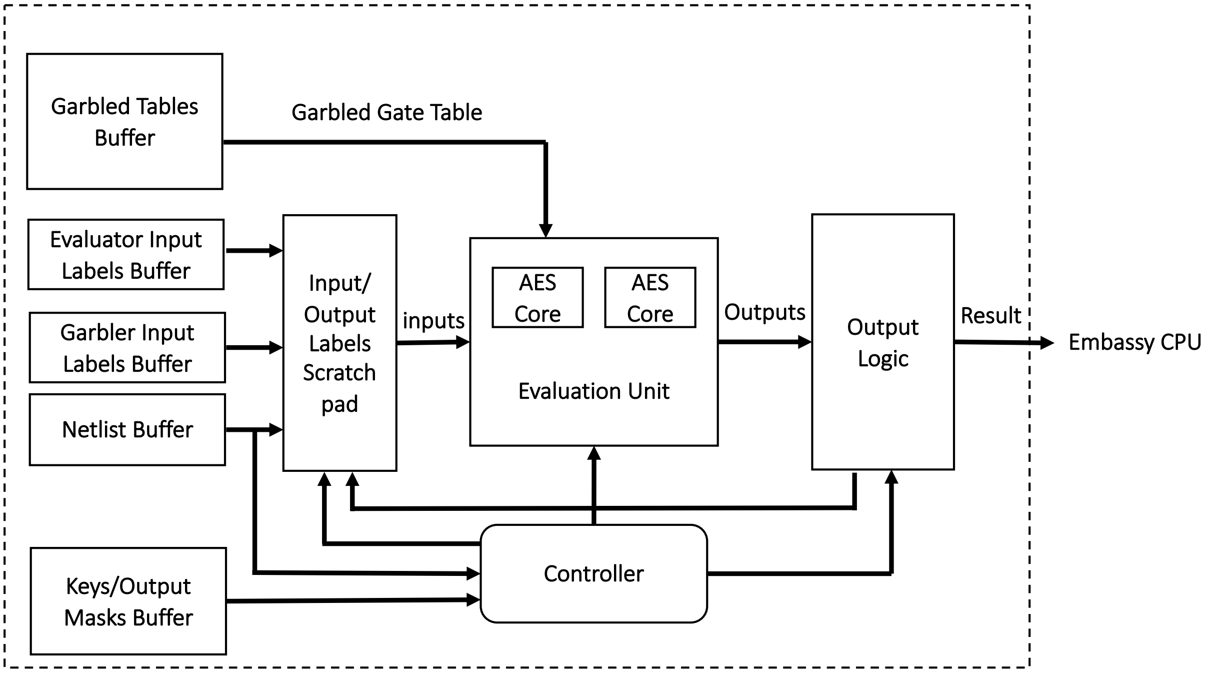


Figure 3.8: Embassy GC Evaluator Architecture. All the inputs stored in the buffers are sent directly from the Garbler except for the Evaluator input labels which are obtained via oblivious transfer (OT) protocol.

3.5.2 The "Ambassador" Garbled Circuit Evaluator Accelerator

While the algorithmic mappings described above take advantage of computational asymmetries, if you make the hardware embassy resource-constrained *enough*, eventually it begins limiting performance again. However, after examining the way these devices are exercised by real code, we observed that *much* of the work is well-structured cryptographic operations amenable to hardware acceleration. We propose that a cryptographic co-processor designed to sit alongside the main Embassy CPU and perform common MPC operations can be used to further improve performance or, more usefully, provide even further computational asymmetry allowing even smaller and more resource-limited devices to be useful in this context. Figure 3.8 details the high level architecture of the

Ambassador Garbled Circuit Evaluator module.

The main component of this module is the evaluation unit which houses 2 AES cores designed to accept one gate per cycle. The garbled tables, garbler/evaluator input labels, and other necessary data such as output masks are obtained from the garbler and are stored in their respective buffer memory. Note that in order to maintain the privacy of input, the evaluator input labels are obtained from the garbler using oblivious transfer (OT). Each pair of input labels is processed in the evaluation unit to obtain the output label which is then sent back to the label scratchpad memory. It will be used in subsequent gate evaluation as the evaluator goes through the netlist gates one by one. Each of the AES core consists of a 10-stage pipeline performing AES-128 on ECB mode. Therefore it improves throughput but introduces potential dependency issues when evaluating the gates whose inputs have not been processed yet. This is the same issue as arises in FASE [64], the project we extended to evaluate the Ambassador. Note that the main operations in Evaluation is the opposite of Garbling where the goal is to use Garbled tables to generate and evaluate a circuit whereas the goal in Garbling is to produce garbled tables. Because of the Half-Gates optimization [20] the amount of work needed to be done by the garbler is 2× more than the evaluator. This explains why our Evaluator unit only needs 2 AES cores instead of 4 to achieve the same throughput performance.

3.6 Evaluation

With the application mapping and inherent algorithmic asymmetry described, the most pressing question is how well a compute-restricted device might actually perform on these MPC applications. Rather than rely on a simulation of the system, we perform direct system experimentation with two machines running the full application stack connected point-to-point. By tuning *down* the performance of the compute and network from this base "1:1" system we can explore the relative impact of network and compute asymmetry on the workload under evaluation.

3.6.1 Methodology

1) Hardware and Software Setup: To simulate the server-Embassy connection, we use two Equinix c3.small [65] bare metal nodes connected with a 10 Gbps LAN. Both machines have an Intel Core E-2278G 3.4 GHz (8C/16T) with 32 GB of memory and a top frequency of 5 GHz. Both are running Ubuntu 18.04.

We emulate a slower machine for the Embassy by scaling down from the maximum operating frequency of the client machine. We achieve this by setting the appropriate *max_perf_pct* Intel p-state parameter that corresponds to the percentage of maximum processor frequency. The particular machine we used for evaluation can be tuned from 800 MHz to 5 GHz (6.25× tuning range). Throughout the evaluation, we make use of 1 GHz (5× slowdown) as our representative Embassy performance. This roughly corresponds to the single-core benchmark performance gap between a typical server-class processor and processor (Intel Celeron N4100) from a commercially-available compute-stick [66].

To accurately simulate a wide sweep of network transfer parameters between Embassy and the server over the LAN connection, we use the Linux *tc* tool. With this tool, we can add artificial delays to simulate latency and throttle bandwidth. We measure the effective network bandwidth and latency using *iperf3* and *ping*, respectively. The default network setting between Embassy and server has an average bandwidth of 9.42 Gbps and an average round trip latency (RTT) of 0.6 ms. We use the available secure neural network implementation from Gazelle [67] and the private set intersection implementations in

Network	Description
SNN-MNIST_NetC	1-Conv, 2-FC, ReLU activation [46]
SNN-MNIST_NetD	2-Conv, 2-FC, ReLU and MaxPool [29]
SNN-CIFAR10	7-Conv, 1-FC, ReLU and MeanPool [29]

Table 3.2: Neural Network Architectures for SNN Workloads

	Intra-Datacenter	WAN
Bandwidth	10 Gbps	200 Mbps
Latency	0.6 ms	40 ms

Table 3.3: Network Parameters

ABY [68] and PSI [69] frameworks. Both applications are written in C++ and were adapted for our Embassy evaluation.

2) Parameter Selection: We consider two network settings: WAN and High-Speed LAN representing the baseline operation and the Embassy operation, respectively. We set the bandwidth/latency configuration for WAN as 200 Mbps/40 ms [70, 71] and High-Speed LAN as 10 Gbps/0.6 ms, which is typical in datacenters. Note that we refrain from selecting extreme network speeds to achieve overly optimistic results although modern datacenters have far more improved network infrastructure reaching bandwidths of 100 Gbps and 400 Gbps [72]. For both applications, we fix the *security parameter* κ to 128 bits. For secure neural network inference, we evaluate the performance overhead of two groups of neural networks designed for MNIST and CIFAR10, respectively. The network architectures are described in Table 3.2. For PSI, we use a 32-bit entry size and fix the number of entries to 100 thousand elements for both client and server, which is a moderate size in DNA matching applications [73]. We include all one-time transfer, offline phase, and online phase costs in our timing measurements. Timing results were averaged over 10 execution iterations.

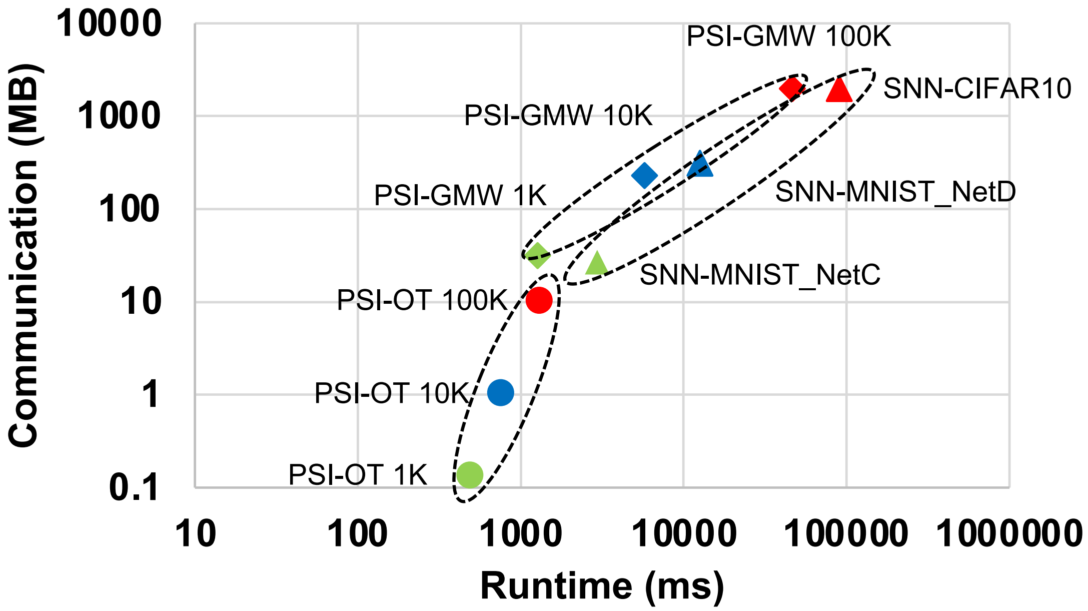


Figure 3.9: Communication volume versus total application runtime. The color-coding indicates the change in input size for each application.

3.6.2 Baseline Embassy Results

Application Communication Cost: Figure 3.9 shows communication volume in MB as a function of the runtime for different applications with various input parameters. We can see that all applications show larger communication costs as the input size increases but they show different characteristics indicated by the slopes of their trend lines. PSI-OT shows little communication and scales well to large input sizes compared to PSI-GMW and SNN. The slope of PSI-OT is also steeper than PSI-GMW indicating that it is less sensitive to communication network improvement. PSI-GMW and SNN have comparable communication that is at least two orders of magnitude more than PSI-OT, while SNN has the highest runtime. PSI-GMW and SNN scale poorly in communication and runtime as input size increases and are thus ideal for Embassy.

Network Bandwidth Limit: Applications can be characterized by their communication-to-computation ratio which is determined by their underlying algorithm and protocol. This property can determine how much performance improvement the application can

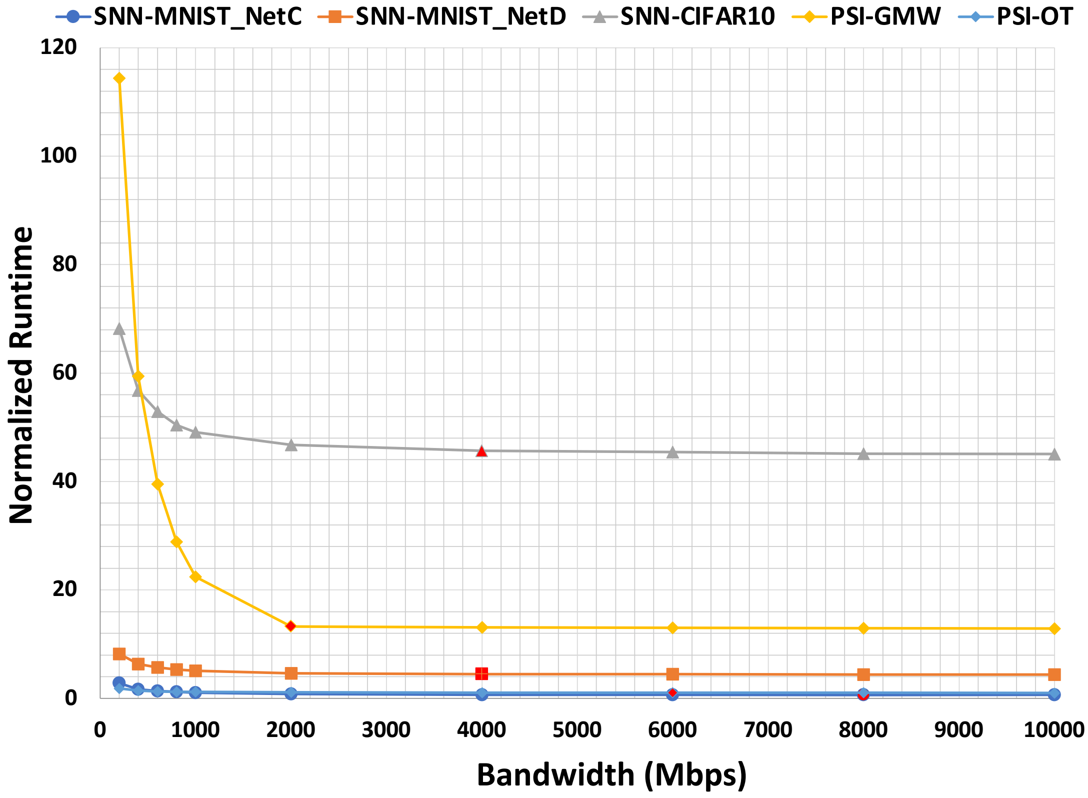


Figure 3.10: Bandwidth limit analysis. Runtime is normalized to the smallest runtime of all applications. Latency is fixed at 0.6 ms (High Speed LAN). The red marker in each line indicates saturation in runtime, where the change in runtime starts to fall below 2% as bandwidth improves.

achieve by improving network bandwidth, i.e., the larger this ratio the larger the potential speedup. Figure 3.10 shows the normalized runtime of the applications for various bandwidth configurations at a High Speed LAN latency of 0.6 ms. PSI-GMW is more sensitive to changes in bandwidth compared to SNN applications below 2 Gbps. Since SNN applications have slightly greater communication-to-computation ratios, they saturate at higher bandwidths (indicated by the red markers in the figure) compared to PSI applications. A key observation is that most applications do not utilize the full bandwidth improvement and become compute-bound before reaching the High-Speed LAN bandwidth. There is no further speedup for SNN-CIFAR10 and SNN-MNIST_NetD after 4000 Mbps. The runtime of PSI-GMW stops decreasing as early as 2000 Mbps. As we will show later in the multithreaded experiments, the reason for relatively low saturation is due to the unoptimized use of threads in the applications. PSI-OT is more dominanted by public-key cryptography computations in base OT and thus shows the least benefit from bandwidth improvement. Note that bandwidth can also be better utilized when we have contention with multiple Embassies in the system.

Network Latency Sweep: Figure 3.11 shows the normalized runtime of the applications for various latency configurations at a High Speed LAN bandwidth of 10 Gbps. Compared to PSI, SNN applications are more sensitive to changes in latency, which are characterized by their greater slopes. This is intuitive because despite Garbled Circuit being a constant-round protocol, a large number of ReLU layers in SNN stacks up communication rounds, while data transfers in PSI can be efficiently batched. The runtimes of all 5 applications scale linearly with latency and show improvement throughout the entire latency range.

Embassy Performance: We illustrate application speedup using Embassy in Figure 3.12 as a function of the performance of the Embassy scaled relative to the server. As discussed in Section 3.6.1, we use core frequency as our performance scaling metric.

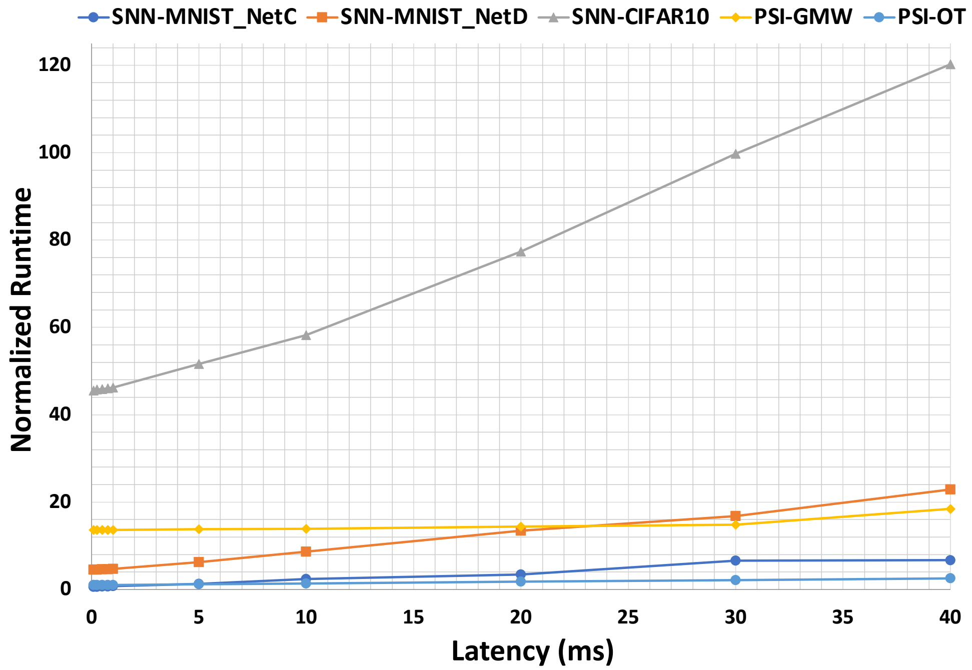


Figure 3.11: Latency sweep analysis. Runtime is normalized based on the smallest runtime of all applications. Bandwidth is fixed at 10 Gbps (High Speed LAN).

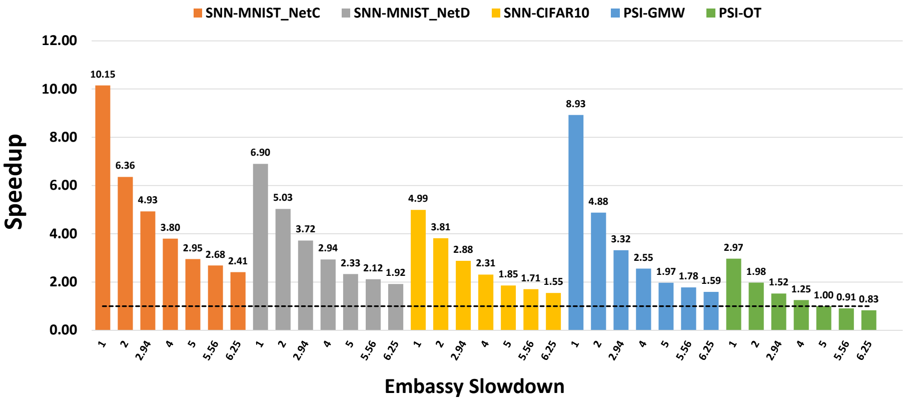


Figure 3.12: Speedup as a function of Embassy slowdown. The runtime speedup for each application is obtained by improving the network bandwidth and latency from a WAN setting (baseline) to the Embassy with High-Speed LAN. The slowdown represents the Embassy performance slowdown relative to a server-class machine. The dotted line shows the slowdown margin for Embassy where the speedup from network improvement is exhausted (speedup = 1).

A speedup of 1 (no speedup) indicates that the Embassy and the server have similar performance and that the speedup is gained from the network improvement from using an intra-system network. This speedup is gradually reduced as the Embassy is slowed down because any benefits from the network are lost from the slower computation. The dotted line represents the *slowdown margin* as this is the point where the speedup from network improvement is exhausted (speedup = 1) from continued Embassy performance slowdown. Note that since in our setup we can only test for a slowdown of 6.25×, we are unable to check the actual slowdown margin for some of the applications.

The slowdown margin of SNN applications is generally higher compared to PSI applications owing to the larger asymmetry in computations (more of the compute-intensive portions of the protocol happen in the untrusted server). For example, for an Embassy that has a slowdown of 5× we can get a speedup of as much as 2.33× in SNN-MNIST_NetD as opposed to 1.97× in PSI-GMW. Shallower networks for MNIST have greater speedup at the same slowdown rate. Within SNN applications, the gap between slowdown margins of the different network architectures comes from the communication composition of the workload. Since SNN-MNIST_NetC uses a significantly shallower neural network compared to SNN-MNIST_NetD and SNN-CIFAR10, communication takes a larger chunk of the overall runtime hence we can get a greater speedup. Note that for PSI-OT there is no speedup at 5× because it has the least communication-to-computation ratio among all applications, meaning that Embassy can barely have any improvement in terms of runtime performance for relatively more compute-bound applications.

Multithreading to Improve Bandwidth Utilization: The previous results show the default unoptimized configuration for the applications with limited thread usage. Since our setup largely alleviates the communication overhead, most applications become compute-bound, as shown in the bandwidth limit evaluation. In Figure 3.13, we illustrate that thread-level parallelism that takes advantage of the available to compute resources

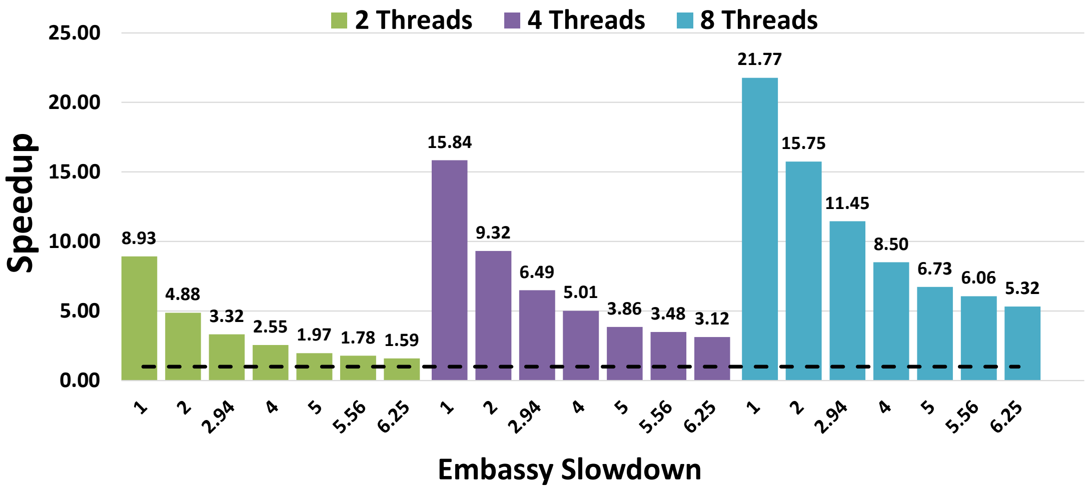


Figure 3.13: Speedup as a function of Embassy slowdown for PSI-GMW with multithreading. The runtime speedup is obtained by improving the network bandwidth and latency from a WAN setting (baseline) to the Embassy setting that uses the High Speed LAN network.

can improve the bandwidth utilization for those applications and in turn the performance of Embassy, which is not possible in the traditional WAN setting [32] with its limited bandwidth. The underlying GMW protocol can be parallelized evenly by dividing the multiplication triple generation in the setup phase to each individual thread [32]. At 5× device slowdown, the speedup grows by 1.96× by increasing from 2 threads to only 4 threads. As the number of threads increases from 2 to 8, the speedup increases by 3.42× from 1.97 to 6.73. Note that reducing the significant overhead of server-side homomorphic encryption in SNN algorithms can achieve a similar effect in the Embassy setting.

Energy Evaluation: One of the key advantages of using Embassy is the energy savings from performing secure multi-party computation locally within a co-located server, because that keeps communication within the co-located server instead of across a WAN. Figure 3.14 shows the estimated energy savings from using Embassy (client-Embassy-server) instead of baseline direct WAN (client-server) computation. Energy consumption is computed as a sum of the total network transfer energy and total computation energy. The network transfer energy gap is conservatively assumed to be 5× [74]. The computa-

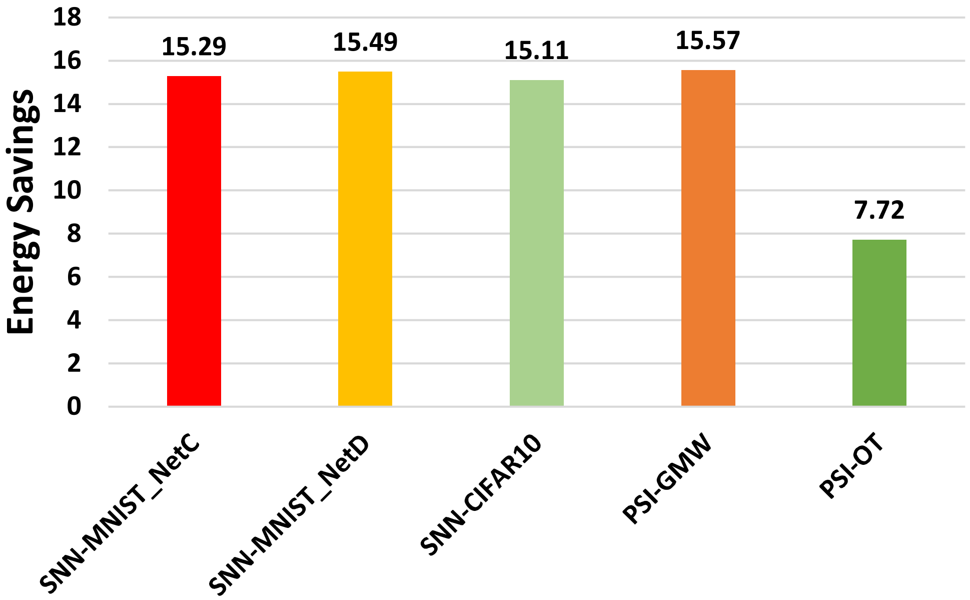


Figure 3.14: Energy savings estimation comparing Embassy to direct WAN computation. TDP is assumed to be 95W for a server and 6W for Embassy. For conventional computation, both client's and server's TDP are the same. The average number of hops is assumed to be 16 end-to-end for WAN while local co-located server hops are assumed to be 5 as for a typical fat tree. Energy consumption is computed as a sum of the network transfer energy and computation energy.

tion energy is computed from a client and server TDP of 95W. Typical energy savings is around 15×. This is mostly due to the use of lower-energy local data movement compared to WANs. Note that Embassy still needs WAN transfers for the client communication but the High-Speed LAN communication still dominates the transfers. For the PSI-OT application, it is less affected by Embassy but 8× savings is still beneficial compared to the WAN setting.

As shown, the power reduction from adding Embassies far outweighs the power increase of the Embassies themselves, because those are more efficient than general-purpose untrusted servers. This is intuitive since the power consumption is 8× per virtual machine (VM) in modern datacenters which span from tens of Watts to hundreds of Watts [75]. Therefore, the additional 6W for an Embassy is comfortably outweighed by less expensive data movement and computation reduction on general-purpose servers.

3.6.3 Ambassador Evaluator Results

An Embassy implemented only as a compute stick-class processor is likely to see a significant performance slowdown as compared to the co-located servers it is connecting to. However, much of the cryptographic calculation that is performed in the MPC setting is amenable to hardware acceleration and so we propose to include such hardware accelerators as part of the Embassy in order to boost both performance and energy efficiency. Here, we investigate the performance improvement available for Embassy if we use dedicated hardware-accelerated implementation of the GC evaluator module to improve GC operations. Our Verilog implementation of the Ambassador evaluator is based on the garbler accelerator provided as part of FASE [64]. We show a comparison of the GC evaluation performance of the Ambassador Evaluator accelerator compared to a system without such an accelerator. Since we are interested in using Embassy for SNN workloads, we focus our evaluation on workloads with non-linear SNN operations like ReLU.

Table 3.4 shows the Ambassador Evaluator's estimated evaluation time and speedup compared to the CPU implementation. We implemented the ReLU circuit shown in Fig. 3.5 in Verilog and obtained the optimized gate count (shown in Table 3.4) from synthesis using Synopsys Design Compiler using the TinyGarble Circuit Synthesis Library [76]. From this gate count, we make use of the similarity in architecture between our evaluator accelerator and the garbler accelerator provided by FASE [64] in order to estimate the expected performance of our evaluator accelerator. We conservatively estimate a range for a processing rate of 2-5.5 cycles/gate based on the simulation results of various circuits reported with the FASE garbler accelerator. We use this cycles/gate to estimate the range of evaluation time and the speedup compared to the software Gazelle implementation of the evaluator. At the FPGA's 100MHz clock frequency, we calculate a performance improvement ranging from 1.57x to 4.31x. Note that even though we demonstrate the advantage of the Ambassador Evaluator accelerator as implemented on FPGA in this study, an ASIC implementation could certainly be used and would result in further improved performance and energy efficiency.

	#XOR	#Non-XOR	#Total	Eval Time	Speedup
ReLU Unit	564	189	753	1506 - 4141 (cc) 15.06 - 41.41 (us)	1.57x -4.31x

Table 3.4: Ambassador Evaluator Performance compared to Gazelle Evaluate function on CPU[30] for the ReLU unit in Fig. 3.5.

Resource Overhead: Our Ambassador Evaluator resource estimation exploits the FPGA infrastructure provided by the FASE Garbler implementation [64]. The FASE Garbler was implemented on a Xilinx Virtex UltraScale VCU108 FPGA while our Evaluator is implemented on a Xilinx Zynq ZCU104 FPGA with lesser system resources. Table 3.5 shows the estimated resource utilization with a clock frequency of 100MHz. As expected, our Ambassador Evaluator accelerator consumes fewer resources than the FASE garbler as it only needs two AES cores compared to the garbler's four.

	Total	%Util
LUT	42472	18.43
Registers	11886	2.58
BRAM	37.5	12.02

Table 3.5: Resource Utilization

Overall SNN Workload Speedup: In order to estimate the overall improvement of introducing a dedicated evaluator accelerator into the Embassy, we profile the SNN applications for the percentage of execution time spent on non-linear layers versus the total runtime. Figure 3.15 shows the percentage of runtime spent on non-linear layers. It shows that the amount of time spent on the total runtime increases when the network becomes deeper and when the network increases in the number of non-linear layers such as ReLU and MaxPool.

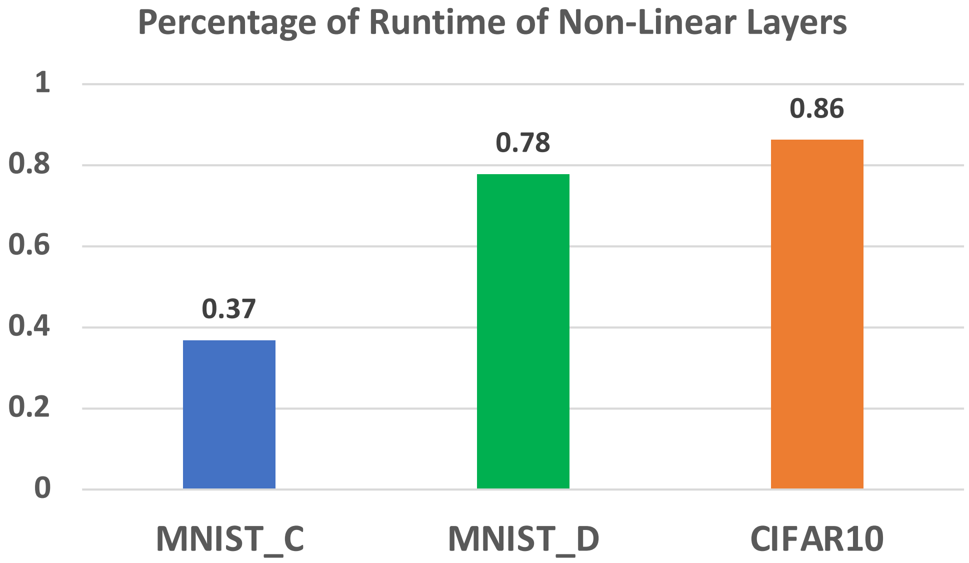


Figure 3.15: Percentage of Non-Linear Layers in SNN Workloads

We use this profiled non-linear execution time and the improvement we obtained from the individual ReLU unit running on the Ambassador Evaluator accelerator to calculate the overall speedup for running the whole network which we observed to range from **1.19**× to **1.52**× in a larger network like CIFAR10. The speedup available is dependent on the type of network and this setup favors deeper networks with more and wider non-linear layers like CIFAR10 as compared to MNIST networks. Note that although it is tempting to think of further speeding up the operation by adding support to Embassy for homomorphic encryption in the linear layers, in the design of Hybrid SNNs, the HE evaluation is done in the server and not in the client/Embassy, thus server support, rather than Embassy support, would be required.

On the Expected Effect of Gate Batch Size. Pipelined garbling and evaluation of the circuit can help improve the performance as well as alleviate limitations on memory capacity since gates that garbled can sent to evaluator directly without waiting for all other gates to be finished. This streaming garbling/evaluation is especially beneficial to Embassy since we are assuming it is equipped with limited memory. An interesting experiment to to find the batch size that can give improvement without significant degradation of as a results of being interactive. Note that being in a datacenter setting allows us to stream the garbling/evaluation more efficiently without significant latency disadvantage.

3.7 Related Work

In considering the viability of our approach in a co-located server setting, we look for inspiration from two trends in co-located server infrastructure design. The first is disaggregated datacenter networks [77], which increase the efficiency and lower the total cost of ownership (TCO) of datacenters using network-attached host-less accelerators. For example, Facebook recently rolled out their F16 [78] datacenter fabric design. The second trend is the adoption of bare metal cloud services [79], where providers allocate dedicated servers for customers. Unlike typical virtual machine-based cloud providers like AWS and Google, doing away with layers of virtualization and dedicating the use of hardware resources results in performance improvements. Further, because clients do not have to share the same physical machines (single tenant), there are fewer potential security risks from recent cross-VM side-channel attacks [58, 59, 60, 25]. Embassies are host-less network-connected computing devices that are exclusively used by clients to perform MPC with co-located untrusted servers.

3.7.1 Trusted Hardware

Trusted hardware such as Intel SGX has been used to support privacy-preserving machine learning [80, 81, 82, 83]. SGX creates enclaves for isolated execution environments and supports remote attestation. However, SGX is fundamentally limited because the trusted execution environment and an untrusted CPU share the same computing resources resulting in a switching overhead. Furthermore, it has limited memory resources (90MB), leading to paging overheads for larger applications [41]. These make these solutions not feasible for evaluating much larger networks, not to mention recent side-channel attacks in SGX [25]. To meet demands for larger workloads, Intel recently released a PCIe interface-based SGX Card with three SGX-equipped CPUs [84]. This hardware with its discrete processors would incur significantly more power than our specialized solution and is a band-aid solution with the same fundamental performance and security flaws of SGX.

Trusted hardware has also been used to support secure multiparty computation. Bahmani et al. [85] make use of code running in the SGX as a trusted third-party and parties which are represented as SGX enclaves perform function evaluation during the online phase. Sartakov et al. [86] extend this by adding support for fast inter-enclave communication. For both works, because of SGX limitations, evaluated applications are very simple such as summations, unlike the applications we consider. Demmler et al. [87] used a trusted secure card in a mobile phone to speed up the generation of multiplication triples in the offline phase of GMW. Our trusted Embassy is a much more capable device that participates in the online phase of the computation. Embassy also physically decouples the computation and does not share any resources with the host. This reduces the number of avenues for side-channel attacks, but does require physical security mechanisms for the Embassy device. Additionally the Embassy device can be flexible in the amount of compute resources it has, allowing the device to be designed to fit the workload.

Bugiel et al. [88] proposed a Twin Clouds model which represents the closest work to our protocol but has many significant differences. First, they make a strong assumption of non-collusion between the two cloud machines. This is not the case for our work since the Embassy is considered a trusted proxy of the client. Second, Twin Clouds' high bandwidth channel is not aimed to improve the network overhead of secure communication but instead, it is used for quick bulk file transfers. Third, they don't describe potential hardware implementation and evaluations.

Eguro et al. [89] proposed FPGA-based secure computation hardware aimed at emulating homomorphic encryption. Our solution, on the other hand, involves no host for the trusted device and can be used to make MPC more efficient.

More recently, Telekine [90] was proposed to mitigate side-channel attacks when clients use cloud-deployed GPUs with TEEs. HETEE [91] was designed to manage all compute units in a server rack by using the PCIe switch fabric to securely allocate accelerators. Unlike Telekine and HETEE, Embassy only considers the security of one single type of portable device.

3.7.2 Secure Computation Cost Characterization

Kerschbaum et al. [92] proposed an automated mixed-protocol selection technique to reduce overall runtime. The secure computation protocols (e.g., homomorphic encryption, garbled circuits, secret sharing) are assigned to primitive operations of the function based on heuristics obtained from a performance model which takes into account factors such as crytographic primitive runtime and network parameters. Pattuk et al. [93] perform similar protocol selection optimization but also takes into account network transmission cost and in general focuses on monetary (dollar) cost reduction on running programs using the secure protocols. von Maltitz et al. [94] study the performance characteristics FRESCO, an MPC framework based on the secret-sharing-based BGW protocol. Unlike other MPC characterization work, they also focus on effect of various network parameters on hardware resources such CPU cycles and memory consumption. Note that, unlike these previous work, our characterization work here is used to evaluate the compute and network transfer cost of two widely-used applications with fixed protocols already in place.

3.7.3 Privacy-Preserving NN Inference

CryptoNets [27] is the first work on privacy-preserving neural network inference we are aware of. It is used as a leveled homomorphic encryption scheme for evaluating all layer which resulted in significant performance overhead and lower accuracy from using square activation functions. DeepSecure [46] used an all garbled circuit approach which improved the computation efficiency of CryptoNets but in turn had worse communication overhead. For example, to perform an MNIST-based inference operation, DeepSecure needs to transfer 791MB per single inference compared to CryptoNets's 595MB for a batch size of 8129. To address this problem, Gazelle [30] proposed a hybrid protocol composed of HE and MPC for NN inference. In this scheme, HE is used for linear operations (e.g., matrix-vector multiplication in convolutional layers) while MPC is used for non-linear operations (ReLU and max pooling functions). This improved the overall compute and communication overheads since HE performs better than GC when the computation has small multiplicative depth (linear function Boolean circuit) and GC is better suited for non-linear functions which can be represented as simple linear-size circuits. However, it still suffers from significant communication overhead because of non-linear layers making it difficult to scale to much larger networks. Our work uses this hybrid protocol for neural network inference but improves on the communication overhead using the Embassy protocol.

XONN [31] proposed the use binarized neural network (BNN) with garbled circuits to speedup linear layers. BNNs use XNOR for multiplication which is considered free when using GC (FreeXOR). This allowed them to make evaluate much larger networks such as VGG. However, as they still use GC for the non-linear layers, there is still significant communication overhead. Furthermore, despite being more efficient computationally, BNNs show significantly lower accuracy. Chameleon [95] proposed the use of a trusted third party to generate multiplication triples during the offline phase. They adopt a seed expansion technique for multiplication triples to save communication at the expense of more computation in random number generation. However, our solution allows the efficient use of the original beaver triple generation with less communication overhead.

There have been proposals to combine GC with other secure computation primitives, such as secret sharing using two untrusted servers, which can be housed by the same cloud and connected in a high bandwidth and low latency channel[45, 96, 97]. These solutions, however, make a strong assumption that two untrusted servers are *non-colluding*.

HEAX [98] proposed the first hardware accelerator implementation for CKKS HE on FPGAs. Cheetah [99] significantly accelerates HE in Gazelle for deeper neural networks by optimizing HE parameters tuning and operator scheduling, while proposing a custom hardware accelerator for server-side HE. The results of HEAX and Cheetah are orthogonal to the contribution of this paper since our solution tackles the communication bottleneck in MPC.

DELPHI [100] improved upon Gazelle by moving expensive cryptographic operations over LHE cipher-texts to the offline phase and proposed to use quadratic polynomials to approximate ReLU, which reduces communication cost. However, DELPHI had to settle with a hybrid approach because of severe accuracy degradation from quadratic approximations.

3.7.4 Tamper-Resistant Hardware

With the rise of mobile and IoT devices, there is greater risk for more sophisticated physical tampering and side-channel attacks. To address this, Google released a tamper-resistant security module [101] used starting from Pixel 2 phones while ARM released Cortex-M35P processor [102] for embedded IoT applications. These solutions can protect against physical penetration and most side-channel attacks (power, timing, electromagnetic).

Two examples of tamper-resistant USB device available in the market are IronKey and Kanguru. IronKey is a FIPS 140-2 Level 3-certified device which zeroizes data or makes the device unusable by applying a wear level current on the device memory after a configurable number of break-in attempts. Kanguru, on the other hand, has a casing that is protected with an epoxy compound, which when removed, destroys the flash chip making the device unusable.

Recently, Immler et al. [53] presented tamper-resistant secure physical enclosure for PCBs. This work allowed for a more practical battery-less physical tampering solution and also proposed the use of PUFs for determining the structural integrity of the device. This tamper resistance mechanism is particularly useful for Embassy.

3.7.5 Hardware Support for MPC

There have been a few works related to hardware support for secure multi-party computation. Songhori et al. proposed TinyGarble [21] to convert big combinational circuits to smaller sequential circuits which is run on multiple clock cycles. The compact circuit results in smaller memory footprint which can fit in the processor cache. As a result, cache misses are minimized during garbling while accessing wire tokens improving garbling performance. This smaller footprint makes it more useful for embedded devices which have limited compute and memory resources.

Implementation and acceleration of the garbling operation have been shown in various hardware platforms [103, 104, 105]. Since they only tackle the issue of GC computation (garbling), overall performance of the protocol is not significantly improved since the bottleneck of the protocol is communication (network transfer) especially with larger applications. This is in contrast to our work which focuses on the communication overhead of secure computation.

3.8 Conclusion

In this paper we explore supporting collections of small but physically secure devices embedded closely with more traditional compute and storage resources. The use of co-located trusted hardware helps resolve the inefficiency of conventional two-party secure computation protocols with surprisingly little compute. We evaluate the ability of such devices to participate in trustworthy computations physically among co-located servers on behalf of a remote client. This general approach could be useful in many different scenarios, but we evaluate one of the most integrated ways one might think to apply such trusted elements: as an active party in a multiparty computation. We show how this Hardware Embassy can leverage a local high bandwidth and low latency network connection to enable more efficient and robust interactive secure computations. We further show that two important privacy-preserving applications, secure neural network inference, and private DNA matching based on Yao's Garbled Circuit (GC) and Goldreich-Micali-Wigderson (GMW), are both amenable to this heterogeneous architecture even without any application specialization. Our experiments indicate that even when the Embassy is 5× slower than external compute resources available, the total system performance is higher due to this increased connectivity. This advantage can be further pressed with addition of specialized hardware, bringing the total performance improvement up to over 4.5×.

Chapter 4

Near-Data Acceleration of Privacy-Preserving Biomarker Search with 3D-Stacked Memory

Homomorphic encryption is a promising technology for enabling various privacy-preserving applications such as secure biomarker search. However, current implementations are not practical due to large performance overheads. A homomorphic encryption scheme has recently been proposed that allows bitwise comparison without the computationally-intensive multiplication and bootstrapping operations. Even so, this scheme still suffers from memory-bound performance bottleneck due to large ciphertext expansion. In this work, we propose *HEGA*, a near-data processing architecture that leverages this scheme with 3D-stacked memory to accelerate privacy-preserving biomarker search. We observe that homomorphic encryption-based search, like other emerging applications, can greatly benefit from the large throughput, capacity, and energy savings of 3D-stacked memory-based near-data processing architectures. Our near-data acceleration solution can speed up biomarker search by 6.3× with 5.7× energy savings compared to an 8-core Intel Xeon processor.

4.1 Introduction

Technological advances in genomic data sequencing has fueled the increased availability of genetic data information and rise of genetic analysis services. Although the abundance of digitized personal genomic information enables advances in bioinformatics and medical domains, it also brings security and privacy concerns. For the analysis of genetic data, there is a growing drive to use privacy-preserving computation techniques to process sensitive genetic information securely [106, 34, 107, 108].

One of the emerging bioinformatics applications is biomarker search. Biomarker search applications are used in medical centers to check for genetic diseases and involve searching a biomarker within a reference database. A match within a reference database indicates high probability of having a certain disease. This type of application was recently explored in the recent iDASH (Integrating Data for Analysis, Anonymization and Sharing) National Center secure genome analysis workshop challenge [109, 110, 111].

Homomorphic encryption (HE) supports operations on encrypted data thus making it possible for data to remain confidential while it is processed in untrusted environments [1, 2]. This property allows for the protection of private data especially in cloud services. HE can be used for secure outsourcing of biomarker search as shown in Figure 4.1. An encrypted biomarker query is sent to a server where it is matched with a database that is also encrypted. The encrypted result (match or no match) is sent back to user where it is decrypted. None of the query, the database entries, and the result is revealed to the server during the entire processing. However, homomorphic encryption has large computational and storage overheads due to large ciphertext explosion [2], thus limiting its practical usage.

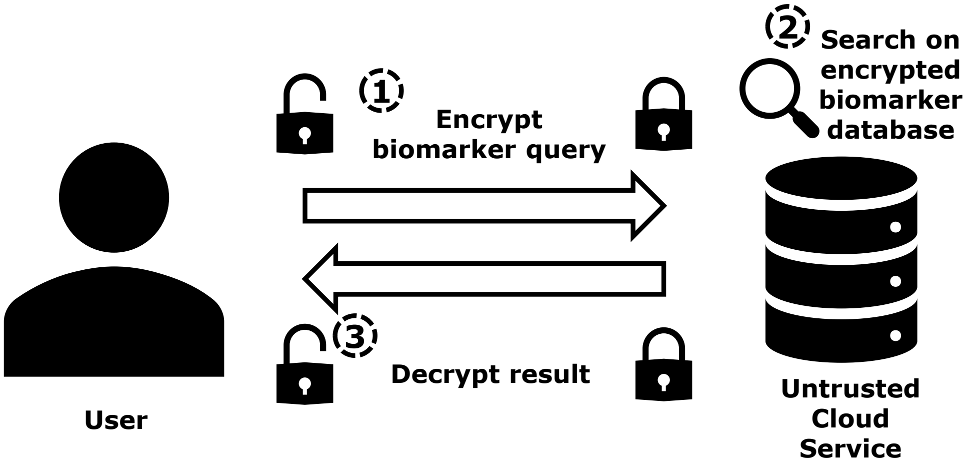

2 Search on encrypted biomarker database
1 Encrypt biomarker query
3
Decrypt result
User
Untrusted Cloud Service

Figure 4.1: Privacy-preserving Biomarker Search Overview

State-of-the-art HE-based privacy-preserving search solutions typically require computationally expensive homomorphic multiplication and bootstrapping operations [112]. For example, homomorphic multiplication typically runs 10-100× slower compared to homomorphic addition, depending on parameters [113]. *Bian et al.* recently proposed an additive homomorphic encryption scheme for exact match search that removes the computational overhead of multiplication and bootstrapping operations [114].

However, a realistic implementation of this scheme still suffers from the large ciphertext explosion of HE which results in heavy data movement during search operation of moving data from memory to the processing unit to perform the comparison. For example, ciphertext expansion for a 32-bit integer for medium security results in 44,000× explosion in size [114]. Therefore, the performance of this scheme is still limited and also not scalable, especially for growing data sizes that require stronger encryption schemes and even larger resulting ciphertexts.

In this paper, we build on top of this additive HE-based search scheme to propose *HEGA*, a near-data processing (NDP) architecture to accelerate privacy-preserving biomarker search. We adopt a 3D-stacked DRAM to reduce data movement and accelerate basic additive homomorphic operation for this application. [1]

[1] Note that although the specific application we explored here is biomarker search, this architecture can also be used

Our contributions in this work are the following:

- We analyze the performance bottleneck of a practical implementation of this homomorphic encryption search scheme
- We propose a 3D-stacked memory-based near-data processing architecture to accelerate search operation based on this homomorphic encryption search scheme
- Using this architecture, we propose the first hardware accelerator for privacy-preserving biomarker search and compare to CPU-based implementation

4.2 Background

4.2.1 Privacy-Preserving Biomarker Search

Biomarker search is one of the key emerging applications in bioinformatics domain [34], as it allows for detection of possible diseases. A specific set of biomarkers are queried from a server that houses a database of these biomarkers. The presence or absence of a specific biomarker or a set of biomarkers indicates a probability of genetic diseases and thus helps medical practitioners to make informed decisions. In dealing with this type of application, however, data is stored in the database and the queries must be encrypted in order to protect privacy.

The biomarkers are stored in Variant Call Format (VCF). These VCF files contain information on biomarkers (genotype information) such as chromosome number and the position of the genome. Furthermore, it contains information for each position such as reference and alternate sequences.

A typical processing flow for HE-based biomarker search is shown in Figure 4.2. The figure shows two general phases: a preprocessing phase and the query phase. In

in other privacy-preserving exact search applications.

the preprocessing phase, each entry in the VCF file is first encoded and hashed before performing the actual homomorphic encryption using a generated key. This is to reduce the size of the encrypted entries since the size of the unencrypted entries will affect the size of the data after encryption. In the query phase, the client similarly needs to preprocess the query before it is sent to the cloud service for the exact search operation. An encrypted result of the search is sent back to client where it can be decrypted using the secret key. In this work, we focus on the homomorphic evaluation stage of the search which takes up the majority of the execution time, especially for large number of queries. For this work, we assume a size of 32 bits for the post-hashed unencrypted database entries and queries. Note that this size is realizable as demonstrated by *Cetin et al.* using a cuckoo-based hashing scheme that enables size reduction of the entries to 29 bits [109].

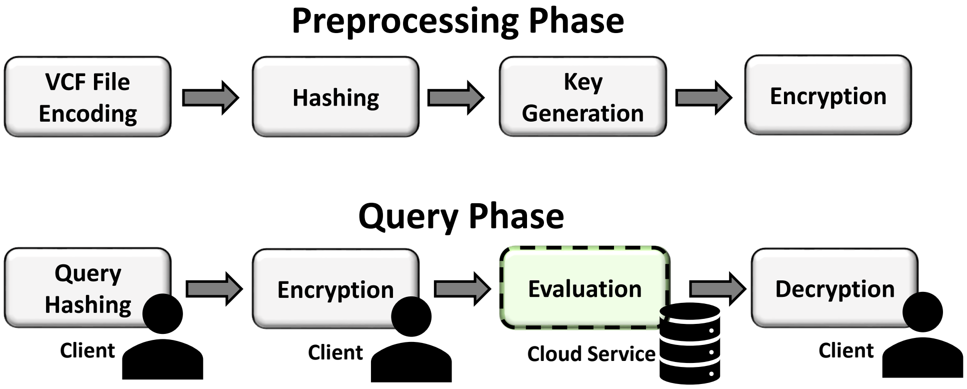


Figure 4.2: HE-based Privacy-Preserving Biomarker Search Flow

4.2.2 Additive Homomorphic Encryption Scheme for Search

Cryptographic solutions such as homomorphic encryption allow for computations on encrypted data. This makes homomorphic encryption a very promising solution for privacy-preserving applications. Fully homomorphic encryption has received wide attention as it allows computations on arbitrarily deep circuits using an operation called bootstrapping [115]. Bootstrapping is a method to refresh a ciphertext by decrypting and re-encrypting to reduce noise, which is a result of performing many HE computations on encrypted data. However, bootstrapping is a computationally expensive operation and thus most recent work on homomorphic encryption also focus on partial (eg. additive) homomorphic encryption schemes.

Although there have been many studies which contributed to the rapid progress of HE, performance bottlenecks continue to hinder its practical realization. Two of the biggest contributors are *data size explosion* and *slow primitive operations*. Encryption results in ciphertext explosion which translates to computation, storage, and communication overheads. Primitive operations such as polynomial multiplication have slow execution times (often millisecond range) and often requires complex specialized hardware [116]. For privacy-preserving search, these problems become even more prominent since aside from the large computational and storage requirements initially demanded by HE. Furthermore, larger HE parameters are needed to support more entries while maintaining the same security level, which exacerbates the data size explosion problem even more.

Ducas et al. proposed the FHEW scheme that can perform NAND operation with only additive homomorphism which greatly reduces the computational requirements[117]. However, it still needs bootstrapping after each homomorphic gate operation which dominates the runtime. More recently, *Bian et al.* [114] proposed SCAM by modifying the plaintext space of FHEW and introducing an encryption constant to implement a two-stage complex homomorphic Boolean gate which can be used for multi-bit word matching. It is also based on additive homomorphism but does not require bootstrapping or multiplication operations which makes it efficient for use in hardware implementations. Equation 4.1 shows a bitwise exact search operation using XNOR-AND gates. SCAM scheme achieves exact search in homomorphic encryption domain using only additive homomorphism in homomorphic XOR-OR gate as shown in Equation 4.2, where c_{x_i} and c_{y_i} are ciphertexts for each bit [114]. In this scheme, each 1-bit plaintext expands to a $(n+1)$ (lg q)-bit ciphertext where q and n are encryption parameters that determined according to the security level. To perform a homomorphic w-bit word matching, $w \cdot (n+1)$ (lg q)-bit integers are added and if the final result decrypts to zero, it means the two words being compared are the same. A non-zero result means the two words do not provide a match.

$$f\left(x, y\right) = \prod_{i=1}^{w} \overline{x_i \oplus y_i} \tag{4.1}$$

$$\text{Hom}\widehat{\text{XOR}}\text{-OR}(x, y) = \sum_{i=0}^{w} (c_{x_i} - c_{y_i}) \tag{4.2}$$

Implementing SCAM for the privacy-preserving search within a database requires performing this search operation in the homomorphic domain through all encrypted database entries and returning the encrypted results of match or no match, with respect to encrypted query bitstream. The client can later decrypt the matching results that evaluate to zero for match and non-zero for no match in the database search. We also adopt the secure two-round communication protocol of SCAM in this work.

This scheme was proposed with an ASIC design [114] in which all encrypted database entries are stored on-chip to provide large bandwidth. However, due to the data explosion of more than 44k× larger data size after encryption, such design results in unacceptable chip area, making it impractical and also not scalable. For example, even for a database of 100K 32-bit entries using their provided encryption paramaters, their ASIC design would already need more than 21 billion transistors, even without including the large on-chip memory (SRAM) required.

We discuss 3D-stacked memories next and in Section 4.3, we discuss why a 3D-stacked memory NDP-based solution is suited for use with this HE scheme and privacy-preserving biomarker search.

4.2.3 3D-Stacked Memories

Hybrid Memory Cube (HMC) is a type of 3D-stacked memory technology which can be used for near-data accelerator architectures. HMC consists of 4/8 DRAM dies on top of a logic base die, resulting 4/8 GB capacity per device[118]. Each DRAM die is divided into 32 partitions, with each partition consisting of multiple banks. Partitions across dies vertically form a vault. Each vault has an independent vault memory controller within the logic die that manages all memory operations for that vault. The logic base die also includes a crossbar switch that connects the vault memory controllers to the I/O ports. HMC uses SerDes I/O links of up to a total of 320 GB/s peak bandwidth. HMC can also be chained together to increase total memory capacity, which can provide a scalable expansion for applications such as privacy-preserving biomarker search which has large memory requirement.

4.3 Motivation

In this section, we discuss our motivation for using a near-data processing approach to accelerate the additive homomorphic encryption scheme and its application in privacy-preserving biomarker search. Performing search using the SCAM scheme is very challenging even though the computing is transformed from complex multiplication into a series of simpler homomorphic additions on the encrypted data, as described in Section 4.2.2. For example, by searching a 10k-entry database that is encrypted with SCAM using encryption parameters in [114], we end up with a slowdown of 60k× on CPU compared to unencrypted operation, which becomes worse for larger databases (75k× for a 20k-entry database), shown in Figure 4.3.

Next, we observe that the application is memory-bound and the challenge lies in performing operations on large data sizes after encryption. Using the same parameters, encrypting the data grows 44k× larger for medium security (80-bit) and 55k× for high security (128-bit) [114]. As a result, a database with 100k entries becomes 16.5GB, which cannot fit on an on-chip cache. At the same time, computation is composed of simple addition operations, making it a memory bandwidth-bound application on all of the available hardware platforms. Using an x86 simulator, we obtain the cycle stack of this application as shown in Figure 4.4. It shows that 72% of the cycles are DRAM-bound stall cycles, which mainly causes the large slowdown of the application.

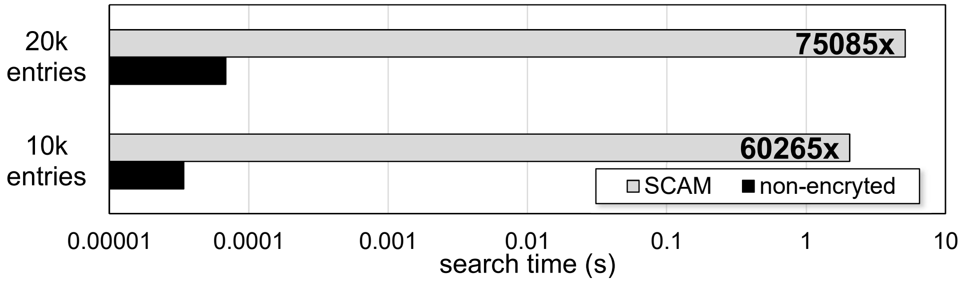


Figure 4.3: Slowdown of database search from homomorphic encryption (SCAM)

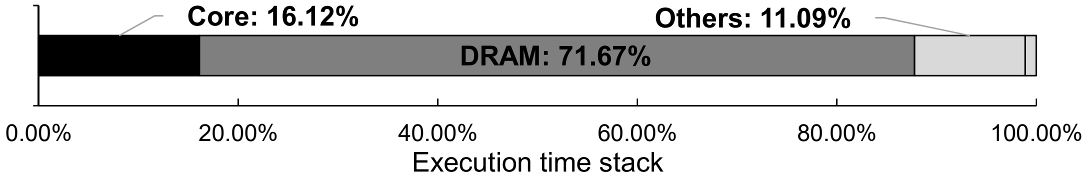


Figure 4.4: Cycle breakdown of SCAM running on CPU+HMC

To further understand this application, we build a roofline model. A roofline model is widely used for high performance computing [119]. The y-axis is the performance (in INT32 ADD), thus the peak computation rate forms the flat part of the roofline. The x-axis is the operational intensity, also called operation/byte ratio, which is a measure of operations per DRAM byte accessed. Applications with higher operational intensity would more likely to be compute-bound, i.e., fall to the flat part of the roofline. Applications with lower operational intensity is likely to be memory-bound (the slanted part of the roofline) and cannot achieve the peak performance of the hardware. We model a SCAM-based database query application where we assume the query data (173KB) is stored on-chip while the database (16.5GB) is off-chip. For each operation (INT32 ADD), we need to fetch 4 bytes of data from off-chip memory, making the operation/byte ratio of this application to be 0.25. We draw the roofline model for various hardware in Figure 4.5, and project the effective performance (indicated by markers) according to the 0.25 operation/byte ratio.

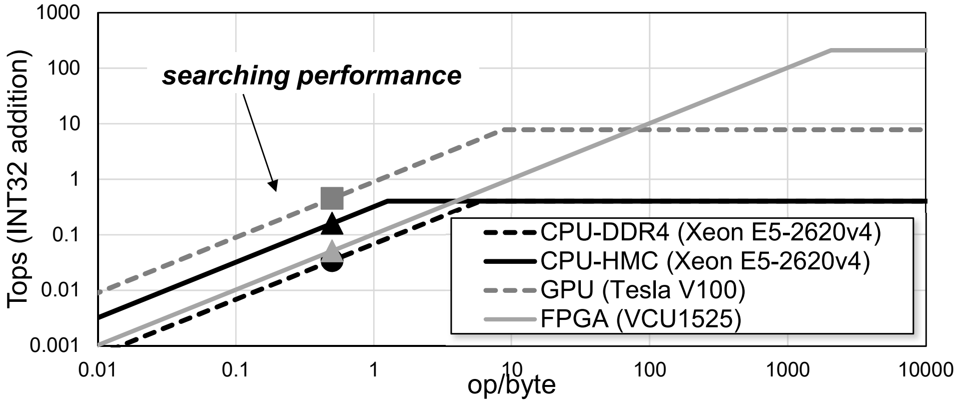


Figure 4.5: A roofline model analysis for SCAM on various platforms (FPGA performance estimation with adder in [120])

We observe that this application is memory-bound for all CPU, GPU, and FPGA platforms, since the small operation/byte ratio falls in the slanted part of these rooflines. We conclude that existing hardware solutions are not suitable or efficient for such application. Specifically, if compared to the CPU-DDR4 with FPGA, although the peak performance is improved from 0.4 TOPs to 221 TOPs, the effective performance only improves 1.5× because the memory bandwidth is not significantly improved (68 GB/s vs. 102 GB/s). On the contrary, the effective performance improves 4.7× with the same CPU but changing from DDR4 (68 GB/s) to HMC (320 GB/s). The simple operations required coupled with the associated large data movement overhead makes it an ideal application to accelerate using 3D-stacked memory where a logic die can be used to implement simple operations. Such architecture can provide massive intra-memory bandwidth and hence solve the memory-bound performance bottleneck. Furthermore, since only simple operations are performed on the logic die using this scheme, it is more suited for 3D-stacked memory integration considering its thermal limitations [121] as compared to typical HE schemes that need complex hardware to speedup the computationally-intensive FFT operation needed in large-integer multiplication.

4.4 HEGA Architecture

4.4.1 *HEGA* Overview

We base the design of our near-memory architecture on Micron's Hybrid Memory Cube [118]. Figure 4.6 illustrates the high-level architecture of our design. The DRAM layers are composed of multiple independent vertical slices called vaults. Each of the vaults can be accessed in parallel, and thus have independent accelerators and memory controllers associated with them. The accelerators can operate on data residing in their local vault and have direct high-bandwidth access to the DRAM layers via the TSVs. The vault controllers handle requests from accelerators co-located within the vault logic, as well as read and write requests that come from the processor.

4.4.2 Architectural Details

Vault logic within the logic die consists of vault memory controller along with the vault processing unit (PU). Each vault PU includes the following components for implementing homomorphic addition, as shown in Figure 4.7. Entry buffer stores the units of a fetched

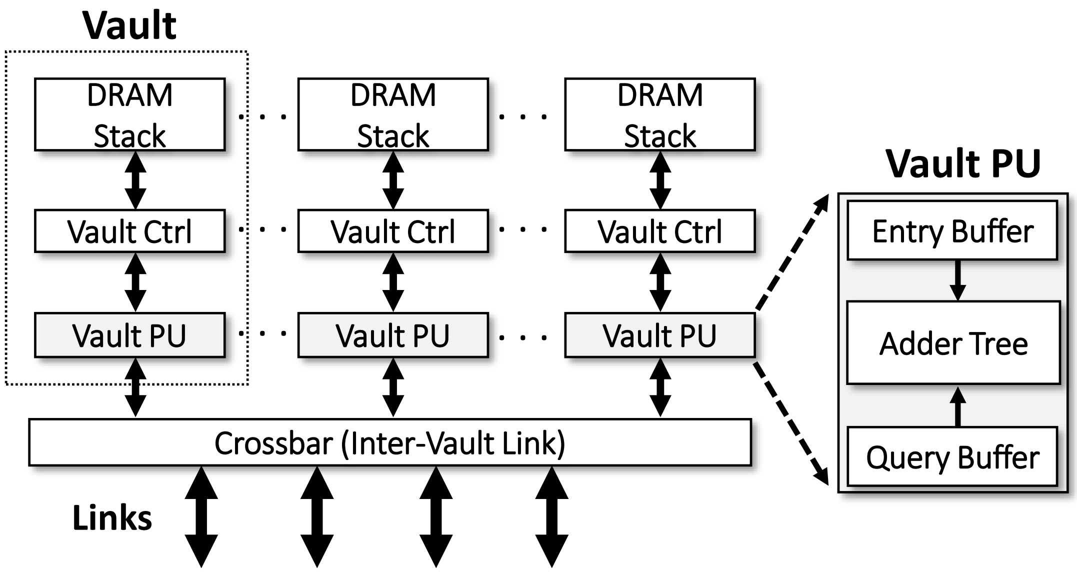


Figure 4.6: *HEGA* Architecture Overview

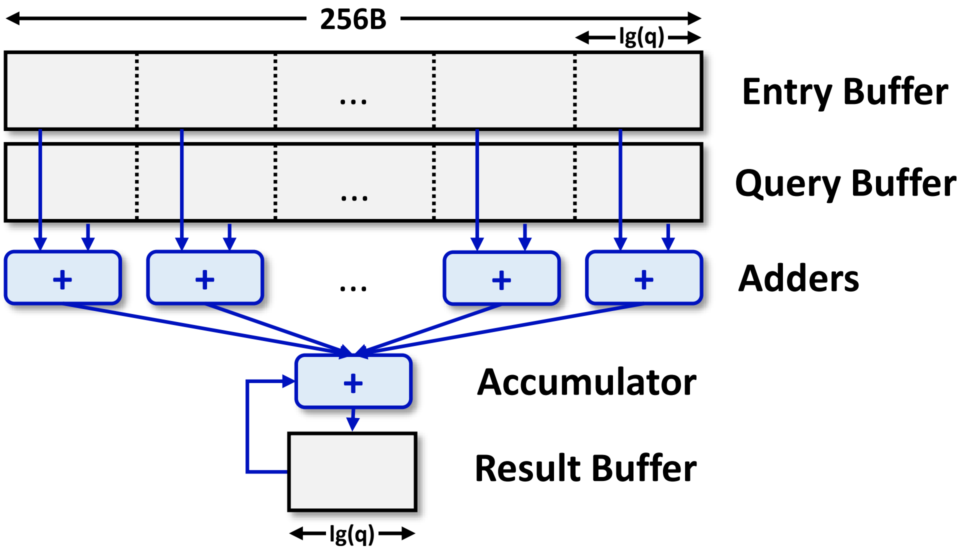


Figure 4.7: *HEGA* Vault PU Architecture (for lg q = 42)

entry from the database. Query buffer stores units of the biomarker query to be searched within the database. Entry and query buffers are 256B in size to match the HMC row buffer size [121]. The adder tree is made of up of adders needed to perform the homomorphic matching as described in Section 4.2.2.

To perform a search operation, a block is first requested to the vault controller and is stored in the entry buffer of the vault PU. Note that the data bus (transfer size) in an HMC vault is 32B and the internal vault bandwidth is defined as 32B/4tCK/vault (10GB/s for tCK = 0.8ns). Once the entry buffers are loaded, the arithmetic units are used to perform (lg q)-bit additions with the partial query data stored in the query buffer. These results are then accumulated and stored in a result buffer. This process continues until all the entry blocks have been processed. The query ciphertext is sent to all vaults to improve efficiency by parallel search. Finally, the search result of size (lg q) bits per entry saved in a result buffer is sent to the user.

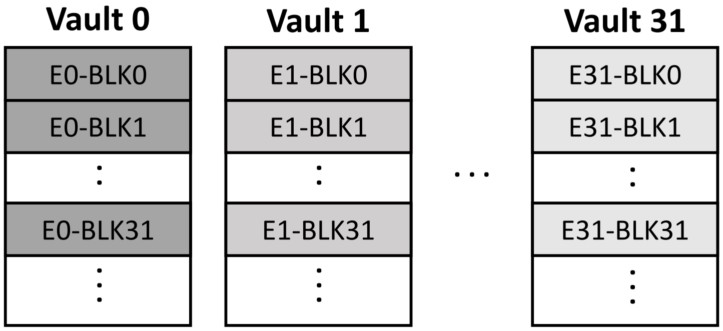


Figure 4.8: HEGA Data Mapping

Next, we discuss mapping of database entries to the 3D-stacked DRAM. To map database entries to the HMC, we use the mapping shown in Figure 4.8. This mapping scheme leverages the vault-level parallelism of the HMC. Each encrypted entry bit composed of $(n + 1)$ (lg q)-bit integers is stored in a vault for the w vaults ($w = 32$). The vault PUs perform the corresponding additions for the entry and query units. Finally, the results from all the vault PUs are accumulated as shown in Figure 4.9. For each entry, $w \cdot (n+1)$ additions of $\lg(q)$-bit entry and query data are computed.

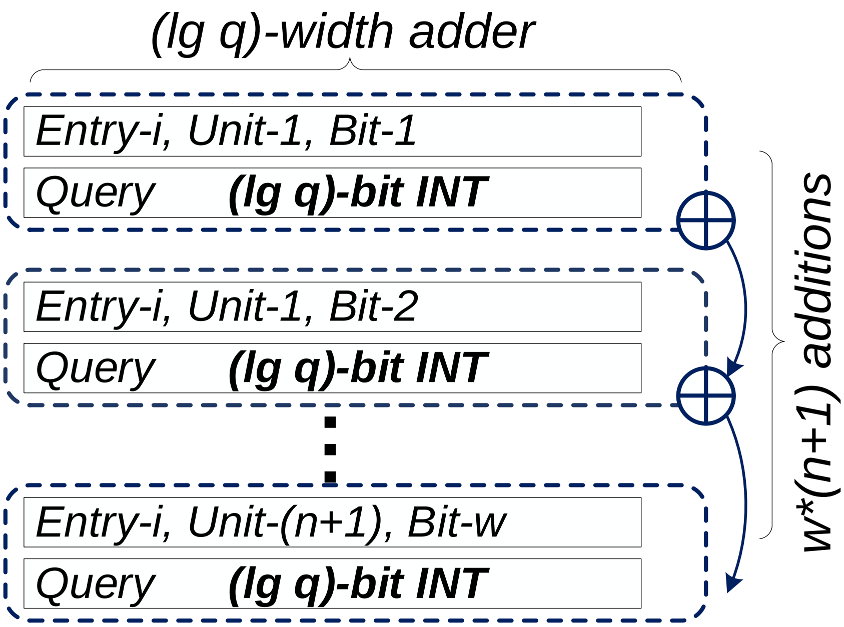


Figure 4.9: SCAM Search Operation

Following the data mapping decribed above, Figure 4.10 shows a sample address mapping from logical binary array address to the HMC physical address using the HE parameters defined in [114].

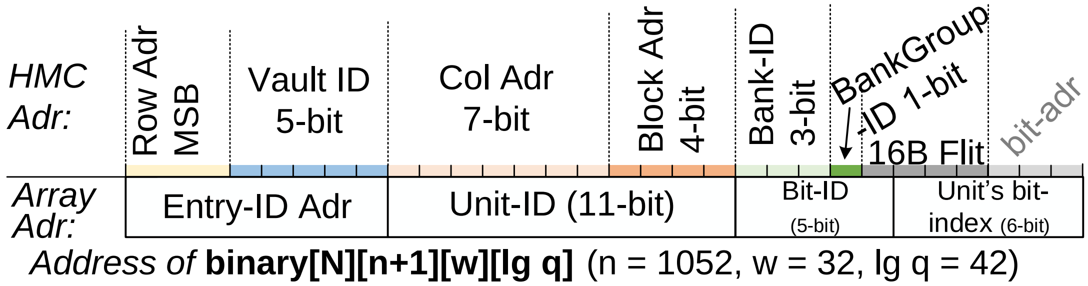


Figure 4.10: HEGA HMC Address Mapping

4.5 Evaluation

4.5.1 Methodology

We use Sniper x86 simulator with custom HMC memory model for our baseline CPU+HMC performance evaluation. The power estimates of the x86 cores were obtained from McPAT integrated in Sniper. We used an in-house simulator to perform *HEGA* performance evaluation. The logic components of our design were synthesized with Design Compiler using NanGate 15nm library. To estimate DRAM energy, we assume a DRAM read energy of 3.76 pJ/bit and a logic layer transfer energy of 6.78 pJ/bit from [122, 123]. Table 4.1 lists the simulation parameters used.

We analyze the following schemes in our experiments:

- **SCAM**: This baseline scheme performs the SCAM scheme on CPU + HMC
- ***HEGA***: Our proposed near-data acceleration architecture which performs SCAM scheme within the logic die of the HMC

Note that to ensure fair comparison, we evaluate SCAM on a CPU + HMC platform and compare to our proposed NDP + HMC platform. We use 32-bit post-hashed unencrypted database entries and queries ($w = 32$) and use parameters $n = 1052$ and (lg q) = 42 as in the instantiation in Table III of [114]. We use a workload consisting of single query on a database of 1k to 16k entries. Note that this small sample range has the advantage of being able to accurately represent the performance of much larger datasets because of the regular workload and at the same time having a feasible simulation speed. Furthermore, this number of entries is big enough to ensure that the size of the encrypted dataset cannot fit into the cache.

Table 4.1: Simulation Parameters

Processor	x86, 8-issue width , out-of-order, 64-entry instruction queue, 2.1GHz, 22nm, 8 cores
Cache	L1D/L1I: 32KB, L2: 256KB, shared L3: 20MB, LRU
HMC	4 links, full-lane, 8GB, 32 Vaults, tCK = 0.8ns, tRCD-tCL-tRP = 17-17-17, tCCDS=4, tCCDL=6
HEGA-Logic (NDP)	32 Vault PUs, 1GHz, 15nm node

4.5.2 Experimental Results

Performance Comparison

Figure 4.11 shows *HEGA* can already provide up to 6× speedup compared to an 8-core Intel Xeon CPU. This shows the limitation of the CPU in utilizing the large bandwidth available in HMC because of its complex cache hierarchy while in *HEGA*, the NDP units can more efficiently use the internal vault bandwidth. Furthermore, even for the small database sizes explored, we observe that CPU performance becomes worse as the size of the database increases, consistent with our observation from Section 4.3.

Furthermore, *HEGA* performs a single word search in $0.61\mu s$ at 1 GHz. For multi-word comparison, *HEGA* leverages vault-level parallelism and pipelining. Although SCAM leverages the parallel structure for fast multi-word search, the ASIC implementation is not realizable for realistic database sizes, as discussed in Section 4.2.2.

Energy Comparison

The normalized energy results are shown in Figure 4.12. Compared to the CPU-based scheme, *HEGA* can reduce the energy by as much as 5.6×. Lower energy for *HEGA* is achieved due to the proximity of data and computation of NDP compared to the CPU, which further allows excluding energy contributions of power-hungry HMC links and crossbar.

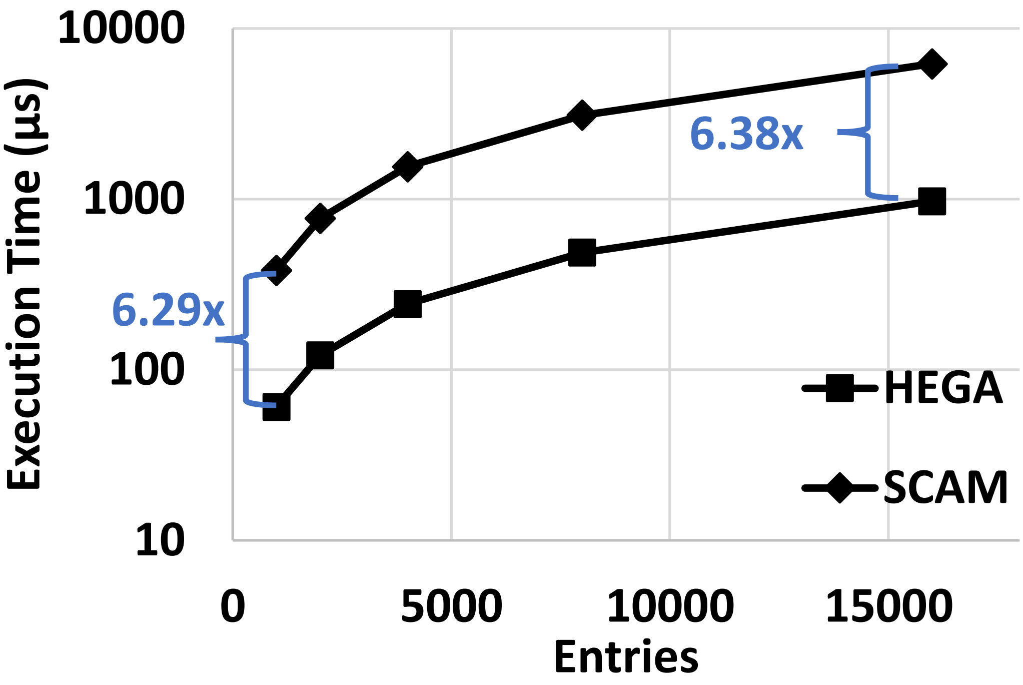


Figure 4.11: Execution Time of SCAM and *HEGA*

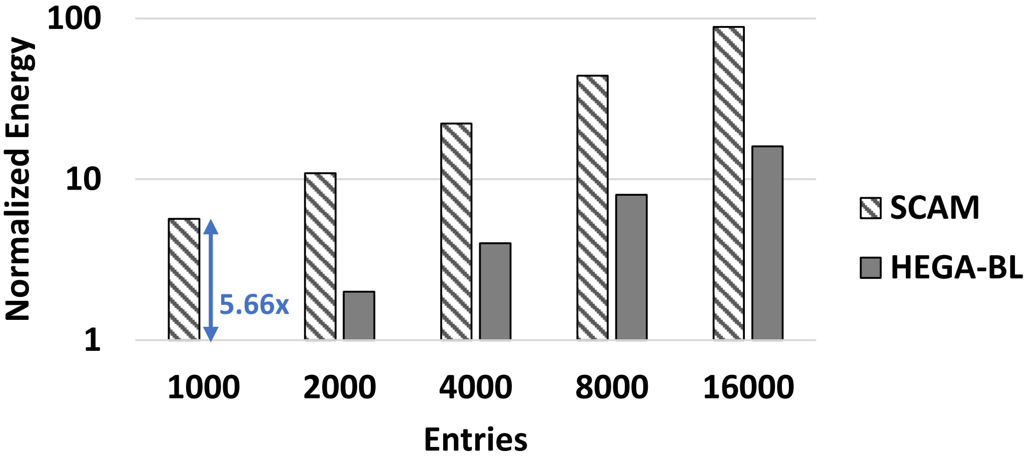


Figure 4.12: Normalized Energy of SCAM and *HEGA*

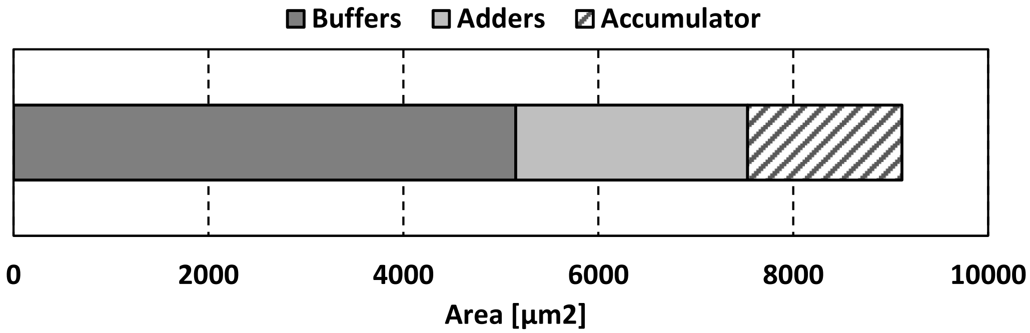


Figure 4.13: Area Breakdown of HEGA

Area Overhead

We obtain the area overhead of *HEGA* in the logic die from synthesis results. Since the vault PUs only include a few simple components such as the buffers and an arithmetic unit, the total area across 32 vaults was calculated to be 0.29 mm^2 (15nm node), which represents just 0.4% area of the HMC logic die [123]. Figure 4.13 shows the area breakdown of main vault PU components implemented in the logic die, namely 256B query and entry buffers, 42-bit adders and accumulator. This evaluation shows that buffers result majority (57%) of area overhead.

4.6 Related Work

4.6.1 Accelerators on 3D-Stacked Memory

Multiple work have proposed near-data architectures using 3D-stacked DRAM to accelerate data intensive operations [124, 125]. *Alves et al.* proposed HIVE [126], an HMC-based architecture which allows performing common vector operations directly inside the HMC. *Kim et al.* proposed GRIM-Filter [127], a near-data processing architecture within the logic layer of a 3D-stacked memory to accelerate read mapping phase of DNA sequencing application.

Even though these work also propose accelerator architectures on 3D-stacked memories, none of these have focused on accelerating homomorphic encryption and its applications such as privacy-preserving biomarker search.

4.6.2 Hardware Acceleration of Privacy-Preserving Search

Few works have presented hardware acceleration schemes for homomorphic encryption-based privacy-preserving search. *Bian et al.* proposed SCAM and an ASIC implementation [114] but has large overheads. *Khedr et al.* introduce a GPU-based approach to homomorphic word searching in their work SHIELD [112]. *Martins et al.* accelerate homomorphic word searching using Intel Xeon Phi [128]. These two implementations still require computationally-expensive homomorphic multiplication. CAMSure [129] allows secure approximate search but biomarker search requires exact search.

Different from prior work on hardware-based secure search, *HEGA* leverages NDP in 3D-stacked memory to handle the large data explosion and the massive data movement due to the streaming search operation. Furthermore, *HEGA*'s use of HMC allows for a scalable solution considering the increasing data expansion rates required for larger databases while maintaining security.

4.7 Conclusion

In this work, we propose *HEGA*, a near-data processing architecture that uses 3D-stacked DRAM to accelerate homomorphic encryption-based biomarker search. We observe that emerging applications like homomorphic encryption-based privacy-preserving search can greatly benefit from the throughput, capacity, and energy savings of 3D-stacked DRAM-based NDP architectures. Our NDP-based solution can speed up search by 6.3× with 5.7× energy savings compared to an 8-core Intel Xeon processor. This work represents a step towards achieving practical homomorphic encryption applications through near-data processing.

Chapter 5

Architectural Modeling of Post-Moore Technologies

With the ever-present push for performance scaling continuing well after traditional device scaling ends, the question remains how emerging devices change architectural tradeoffs as well as what is each device's long-term impact potential. Many of these emerging devices are still being studied in device-level and the actual potential benefits or disadvantages of these technologies are still typically not clear because of the lack of system-level studies. In this work, we show how high-level modeling and analysis can be used to explore the system-level potential benefits and limitations of emerging technologies.

We divide this chapter as follows. First, we discuss modeling and quantifying the advantage of Post-Moore technologies in domain-specific accelerators. Specifically, we present a case study analyzing the how an accelerator based on NTT (Number Theoretic Transform), a common module used in many Post-Quantum Cryptography schemes, can be rapidly modeled and evaluated using Post-Moore technologies. Next, we discuss another Post-Moore option, superconducting electronics technology, could be modeled as a multicore processor. Additionally, we propose and evaluate technology heterogenuity represented by a hybrid configuration for this emerging technology.

5.1 Post-Moore Technologies in Domain-Specific Accelerator Designs

Specializing chips using hardware accelerators has become the prime means to alleviate the gap between the growing computational demands and the stagnating transistor budgets caused by the slowdown of CMOS scaling. Specialized hardware accelerators have filled the performance improvement despite slowing CMOS scaling. However, there is still a limit on performance gains coming from chip specialization (”accelerator wall”), making it harder to satisfy more increasingly demanding workloads. An increasing number of emerging technologies have been proposed to replace CMOS technology, however, there has been surprisingly no straight forward way of evaluating these technologies for accelerator applications.

In this work, we show a rapid evaluation method for Post-Moore-based accelerators. This work enables rapid exploration of the system-level impact of emerging technologies making it easy even for technologists to decide on which promising technology use for certain applications. To demonstrate our framework, we present an analysis of Polynomial NTT Accelerator which can be used in Kyber, a type of Post-Quantum Cryptography KEM, and many other LWE-based cryptographic modules used in datacenter privacy-preserving applications.

5.1.1 Introduction

Recent work have shown that improvements in architecture will not be enough to sustain continuous improvement in performance [8], which implies that parallel improvements in technology would still be ultimately be required. And as Denard Scaling and Moore's Law have been shown to have its limits, it is important to understand emerging technologies and how they can be used to replace silicon in future processors and accelerators. However, most emerging technologies are usually studied in low level (device and circuit level) analysis. Doing this, it is often difficult to know the real impact of these technologies for particular applications which can only be evaluated at the system level. In particular, there is no rapid and systematic way of evaluating whether a new technology can be a good candidate for a new accelerator architecture for a particular application. On the other hand, there have been large number of tools and techniques for evaluating accelerators but they lack support for emerging technologies. In this work, we bridge this gap by proposing a framework for rapid evaluation and modeling of system level architecture of Post-Moore-based accelerators.

5.1.2 General Flow for Rapid Evaluation of Post-Moore Accelerators

Our proposed flow is shown in Figure 5.1 which is based from the Aladdin [130, 131] framework. Aladdin provides specific hard-coded values for a particular commercial Si CMOS library (40nm) and does not support other technologies out of the box. We extend Aladdin by allowing easy integration with other technology libraries. When the generating the tech file is a new technology is challenging, we propose estimation of required the functional unit parameters needed in the Aladdin simulation from smaller components as we will describe in the next section.

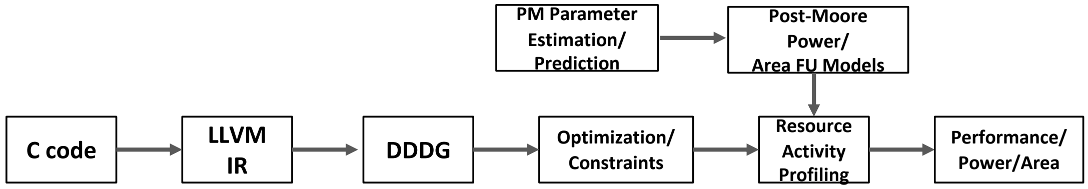

PM Parameter Estimation/ Prediction
Post-Moore Power/ Area FU Models
C code
LLVM IR
DDDG
Optimization/ Constraints
Resource Activity Profiling
Performance/ Power/Area

Figure 5.1: Proposed General Flow for Rapid Design Space Exploration for Post–Moore Technology-based Accelerators which is based from Aladdin [130]

5.1.3 Statistical Estimation of Post-Moore Library Parameters

Standard cell libraries are typically derived from Verilog device models and template library in a process called characterization. However, for emerging technologies, it might be difficult to perform characterization. Therefore, we propose a statistical estimation method to estimate the parameters of the PM library.

When trained for a RTL of a small functional units, we propose to estimate parameters such as area, delay, and power consumption of large functional units using linear programming-based optimization. First we decompose the RTL into a set of logic-complete gates such as AND/OR/INV. The predicted parameter (PP) would then be equal to

$$\mathrm{PP} = \sum_{i=1}^{p} n_i \times k_i \tag{5.1}$$

where n_i and k_i are refers to the numbers of gates and the coefficients of the gates for a set of p logic-complete gate set. The values for these coefficients are then obtained by solving the minimization problem of the form

$$\min \sum_{i=1}^{r} |\mathrm{AP} - \mathrm{PP}|^2 \tag{5.2}$$

where AP refers to the actual value of the parameter and r is the number of subunits used for training. The coefficient k_i are constrained to be positive. Note that in many emerging technologies, it would be suitable to design small subset of functional units which can then be used to estimate larger units so this r can be small. A similar approach has been used in [132] to estimate the parameters of from VHDL RTL descriptions but our approach also extends to power, has a different objective function, and uses this in the context of estimating parameters from emerging technologies with are more challenging to characterize and get library files for.

We test this technique for a set of both CMOS and Post-Moore technology libraries. For this work, we focus on NCFET as a representative technology but other Post-Moore devices/libraries should be applicable. For training, we used small subset of ADD, MULT and BITWISE gates with bitwidths up to 4. From this, we estimate the set of functional units required for the Aladdin-based simulation, as well as other larger cryptographic units and compare with actual synthesis results. We obtain the actual parameters values from synthesis with Synopsis Design Compiler using NanGate 15nm and ASAP 7nm-based NCFET tech libraries [133]. The required functional units in Aladdin are shown in Table 5.1.3. For each parameter and technology set, we use Gurobi [134] to solve the coefficients and use those coefficients for estimation. Figure 5.2 and Figure 5.3 show the results of the estimation of area and power, respectively. For area estimation in Figure 5.2, they fall generally under 25% and this estimation has worse results with smaller functional units and smaller technology nodes as expected because of the larger effect of over/under estimation on the smaller area magnitudes. For power estimation in Figure 5.3, the estimation is typically under 30% except for some large functional units. Note that the power estimation we used for comparison comes from Design Compiler which assumes a fixed switching activity. Therefore, more accurate power estimation can be done by testing different activity factors.

Unit	Description
ADD	Adder
MULT	Multiplier
BIT	Bitwise-Operation
SHIFTER	Shifter
REG	Register
FP_SP_ADD	Single Precision Floating Point Adder
FP_SP_MULT	Single Precision Floating Point Multiplier
FP_DP_ADD	Double Precision Floating Point Adder
FP_DP_MULT	Double Precision Floating Point Multiplier
FP_TRIG	Single Precision Floating Trig Function

Table 5.1: Aladdin Functional Units

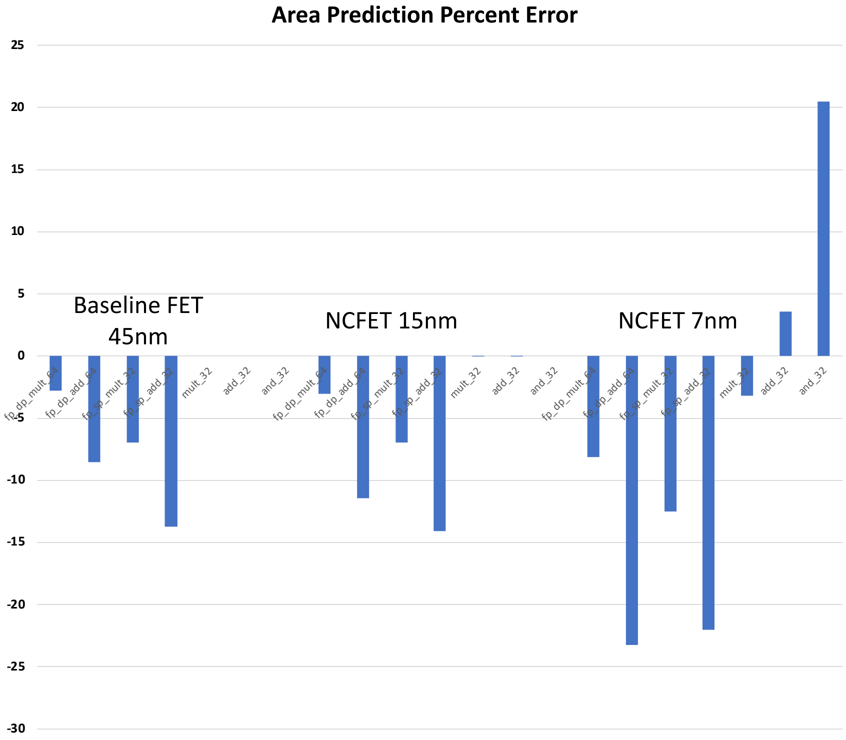


Figure 5.2: Comparison of predicted and actual (synthesis) results for selected functional units

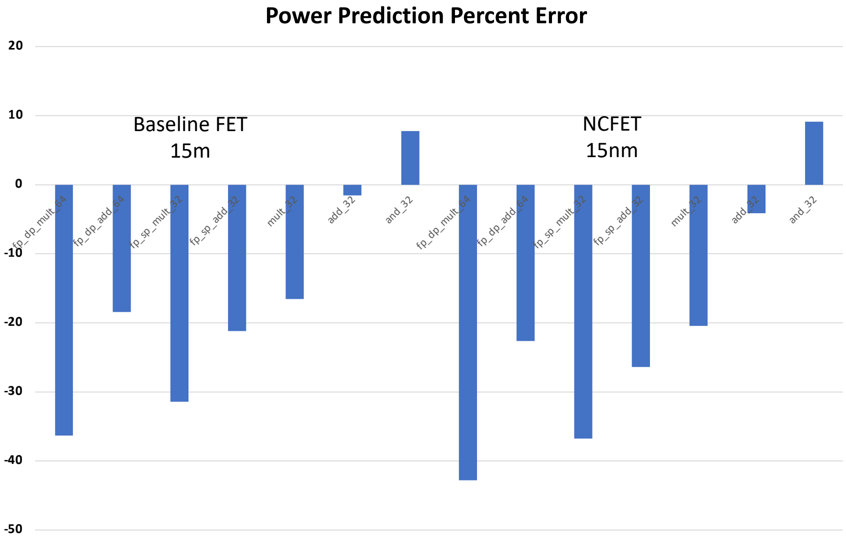


Figure 5.3: Comparison of predicted and actual (synthesis) results for selected functional units

5.1.4 Post-Quantum Cryptographic Accelerator

While using Post-Moore technologies for can be used for various hardware accelerators, in this work, as a proof of concept, we focus on the use of Post-Moore technologies for the NTT-based polynomial operation which is used in many Post-Quantum Cryptography schemes like Kyber. Kyber is a KEM (Key Encapsulation Mechanism) that is designed to be used in cryptographic applications. It is a low-complexity, high-security, and fast cryptographic protocol that is designed to be used in the context of secure communication. KEMs allows for encapsulation of symmetric key by using asymmetric cryptography. The required inputs and outputs of the encapsulation and decapsulation operations are shown in Figure 5.4. Kyber has recently been selected by NIST for standardization after three rounds of selection process [135]. Kyber is based on LWE-based encryption, specifically, Module-LWE (MLWE). To have better support for current hardware, a variant of Kyber was also proposed called Kyber-90s, which uses AES-256 in counter mode and SHA2 to replace SHAKE. Table 5.2 shows the parameter set for the Kyber, similar to various parameters for AES and SHA, i.e., Kyber512 was targeted to

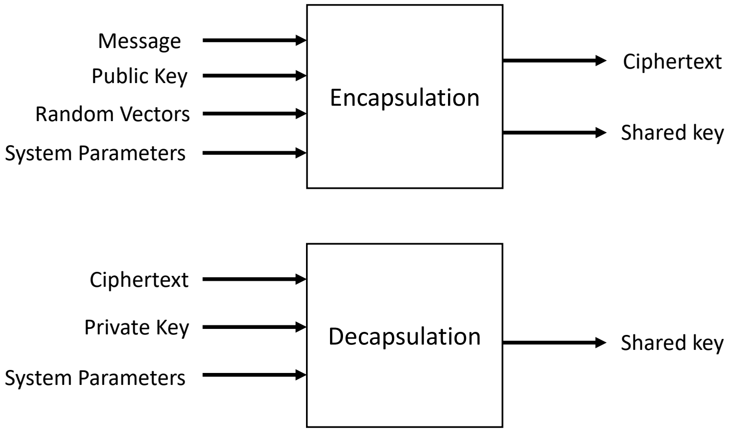


Figure 5.4: Encapsulation and Decapsulation Process in Post-Quantum Cryptography KEMs

	NIST Level	Quantum Hardness (bits)	Parameters					Size (bytes)		
			n	k	q	(η_1, η_2)	(d_1, d_2)	Secret Key (sk)	Public Key (pk)	Ciphertext (ct)
Kyber512	1 (AES128)	102	256	2	3, 329	(3, 2)	(10, 3)	1632	800	768
Kyber768	3 (AES192)	161	256	3	3, 329	(2, 2)	(10, 4)	2400	1184	1088
Kyber1024	5 (AES256)	241	256	4	3, 329	(2, 2)	(11, 5)	3168	1568	1568

Table 5.2: Comparison of KYBER Parameter Sets [137, 136]

have similar security as AES128, Kyber768 with AES192, and Kyber1024 with AES256. The authors of Kyber recommend the use of Kyber768 parameters as it gives more than 128 bits of security against all known classical and quantum attacks [136]. In this work we focus on Kyber768 which we will refer to as just Kyber from hereon.

Profiling

Table 5.3 shows the profiling results Kyber using a clean reference implementation [138]. The profiling is obtained from running hotspot mode in $perf$. Here, it clearly shows that hashing operation Keccak dominates the runtime along with NTT/Inverse NTT and the reduction operations which are used for in polynomial multiplication. Keccak is also used by another Module-LWR based post-quantum cryptography KEM such as SABER [139], as shown in the profiling results in Table 5.4 and however, it uses Karatsuba method for polynomial multiplication. Thus, polynomial multiplication and Keccak are prime targets for hardware acceleration. For Kyber NTT-based polynomial multipli-

Unit	%
Keccak	33.1
Montgomery Reduction	16.36
Inv NTT	8.68
Barrett Reduction	8.51
Base Mult	7.72
NTT	5.1
Gen Matrix	3.73
Others	16.8

Table 5.3: Kyber Encapsulation Function Profiling Results

Unit	%
Karatsuba	46.52
Keccak	30.20
Poly Mul Acc	10.22
BS2POLVECq	3.76
Shake128	2.26
Others	7.04

Table 5.4: SABER Profiling Result

cation, it uses the Cooley-Tuckey algorithm for forward NTT and the Gentleman-Sande algorithm for inverse NTT.

5.1.5 Methodology

As profiling of some Post-Quantum cryptography shows, they are heavily dependent on polynomial multiplication and hashing functions. This is consistent with results from earlier related work which focuses on general module-based synthesis of accelerators for a wide variety of R-LWE Post-Quantum Cryptography [140]. Note that our work is different in that we target different emerging technologies and for this demonstration, we focus on acceleration of polynomial multiplication. Specifically, we focus on enabling fast polynomial multiplication using Polynomial NTT which involves NTT, Barrett Reduction, and Montgomery Reduction which make up significant portions of the software runtime of Kyber. We use the Kyber code from provided from PQClean [138] and extracted the Polynomial NTT function. We annotate critical loops of the code which can be unrolled and accelerated. We synthesize various functional units using 15nm and 7nm Baseline FET and NCFET technologies. We then modify Aladdin to make use of these power/area/delay values to estimate parameters of the cache-based accelerator. We use a cache-based accelerator architecture as shown in Figure 5.5. The accelerator is connected to the system bus and and it has it's own cache-based memory. Aladdin provides several knobs for design-space exploration of accelerators. In this work, as a proof-of-concept, we focus on the number of lanes of the accelerator which represents the unrolling factor of loops in the program. The larger the number of lanes, the more hardware resources needed but the faster the accelerator will run. We compare Baseline FET and NCFET for both 15nm and 7nm nodes. For 15nm, Baseline FET has VDD of 0.8V while NCFET has VDD of 0.4V while in 7nm, Baseline FET has VDD of 0.7V while NCFET has VDD of 0.5V. As explained in the earlier section, due to the lower VDD, NCFET can achieve lower power consumption but with no significant impact on performance. In the following comparisons, we report the only the parameters of the functional unit and not include memory. Estimating the memory parameters based on Post-Moore technologies is an interesting future work we are considering. To measure the performance improvement of using an accelerator, we use the reported gem5 statistics and compare the number of cycles to complete the operation with and without using the accelerator.

5.1.6 Analysis of NCFET-based Polynomial NTT Accelerator

Figure 5.6 shows the decreasing number of cycles required of the Polynomial NTT Accelerator for increasing number of lanes (loop unrolling factor) as expected. For this design, the number of cycles saturates at around 4 lanes. With this information, we focus

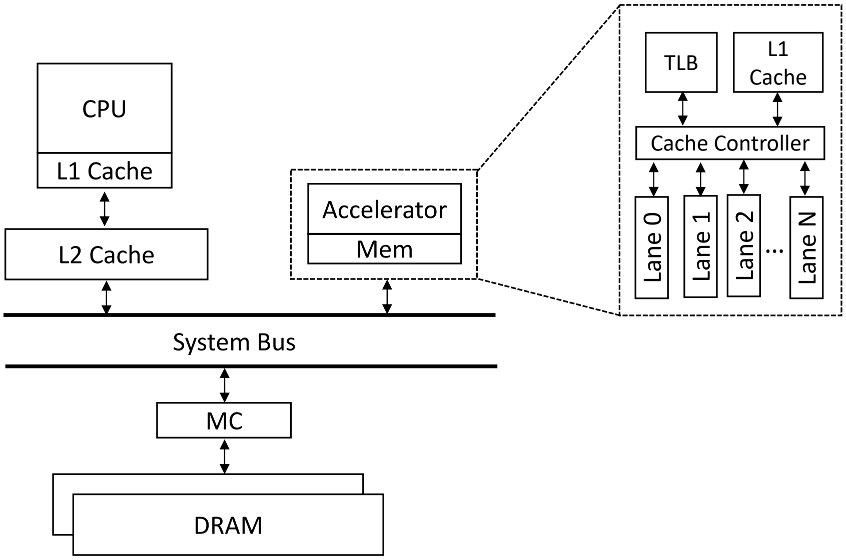


Figure 5.5: Architecture of a Polynomial NTT Cache-based Accelerator

on the comparing a single lane and 4 lanes of the accelerator. Figure 5.7 and Figure 5.8 show the difference of power consumption of the Baseline FET and NCFET for 7nm and 15nm technology, respectively. For 15nm, NCFET provides around 6.3× power reduction compared to Baseline FET. However, this reduced to around 1.7× power reduction for 7nm technology. One reason for this the gap of the VDDs used is much larger in the 15nm case. As we will show later, the delay in 7nm for NCFET is already worse with the current VDDs used compared to Baseline FET, and lowering the VDD even more, will result of course result in more power reduction but at worse performance. For both of these results, the number of lanes does not significantly impact the difference between Baseline FET and NCFET power. Using this 7nm NCFET-based accelerator and a host processor running with a clock of 2000MHz, we can achieve a performance speedup of around 1.27× compared to a system without the accelerator. With an accelerator with 4 lanes, the operation runs in 342703 cycles while without an accelerator, it runs in 436744 cycles.

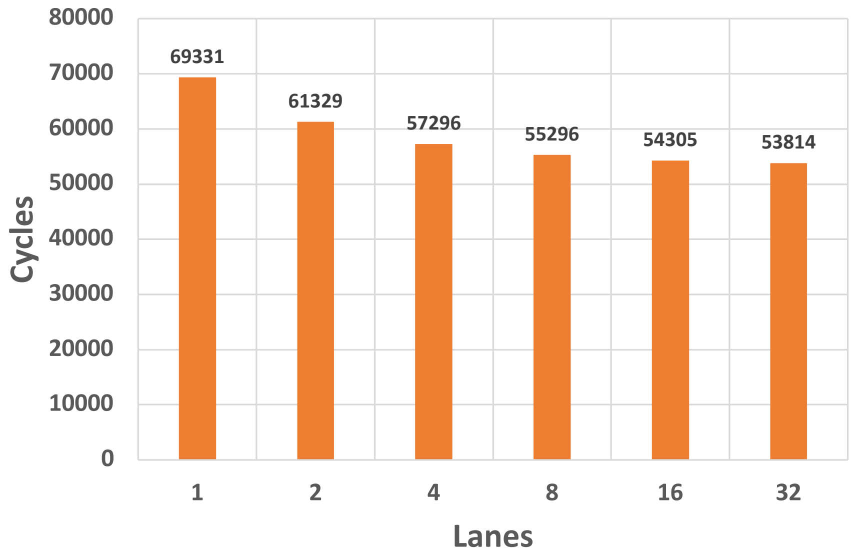


Figure 5.6: Accelerator cycles as a function of the number of accelerator lanes.

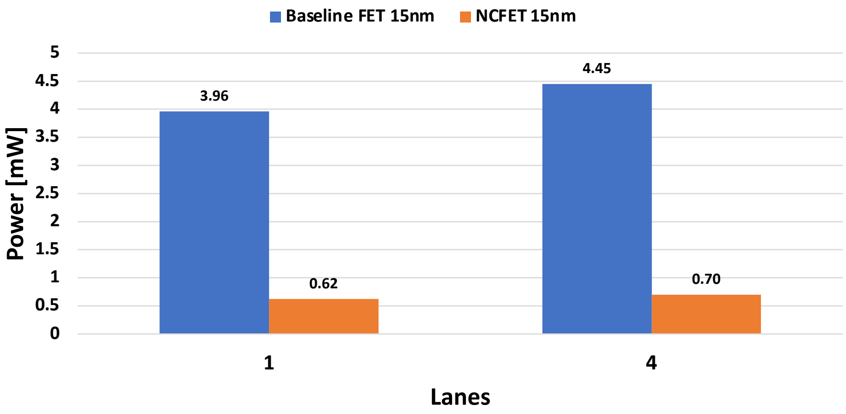


Figure 5.7: Comparison of Power between Baseline FET 15nm and NCFET 15nm with 1 and 4 accelerator lanes (loop unrolling factor)

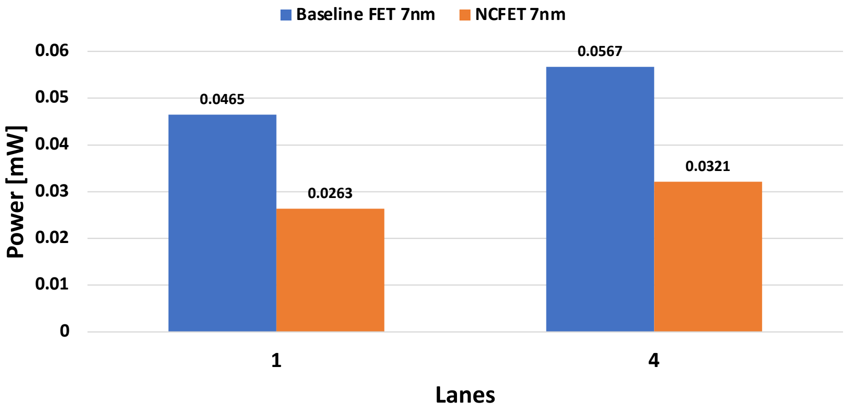


Figure 5.8: Comparison of Power between Baseline FET 7nm and NCFET 7nm with 1 and 4 accelerator lanes (loop unrolling factor)

5.1.7 Analysis of NCFET for Cryptographic Modules

We also compare NCFET libraries with corresponding Baseline FET libraries for other cryptographic modules. Specifically, we look at AES256, SHA256, KECCAK and 517x517 3-stage Karatsuba Large Integer Multiplier which are typically used in LWE-based Post-Quantum Cryptography. The results of the synthesis are shown in Table 5.1.7. For area, since the NCFET libraries are based from the Basline FET, there is no significant difference. For both 15nm and 7nm technologies, NCFET, despite having lower VDD compared to Baseline FET, has similar delay (performance) but offers significant power reduction. Power consumption is largely dominated by internal power. The total power gap in the 7nm node is around 1.8× while in the 15nm node, the power gap is around 4.5×. This implies that the power savings advantage of this technology is reduced as we go to smaller technology nodes.

5.2 Modeling a Hybrid Superconducting Processor

Superconducting electronics offer the promising prospect of high performance (tens of GHz) at low switching energy (attojoule level). However, this new type of computing

	Crypto Module	Technology	VDD	Area	Switching	Internal	Leakage	Total Power	Delay
7nm	AES256	Baseline FET 7nm	0.7	2477.71061	1.02E-02	0.671	1.71E+03	0.683	2.33312
		NCFET 7nm	0.5	2476.44215	5.97E-03	0.371	8.94E+02	0.378	2.80518
	SHA256	Baseline FET 7nm	0.7	1377.50381	8.34E-03	0.408	9.84E+02	0.418	3.10853
		NCFET 7nm	0.5	1419.46505	4.92E-03	0.226	5.32E+02	0.231	3.96485
	KECCAK	Baseline FET 7nm	0.7	2743.5769	5.64E-02	0.731	1.82E+03	0.789	1.98314
		NCFET 7nm	0.5	2756.08654	3.47E-02	0.405	9.42E+02	0.441	2.36584
	Karatsuba Multiplier	Baseline FET 7nm	0.7	5783.15698	0.354	1.672	3.87E+03	2.03	3.98448
		NCFET 7nm	0.5	5808.55534	0.213	0.927	2.03E+03	1.142	3.98726
15nm	AES256	Baseline FET 15nm	0.8	8754.41349	5.12E-02	2.883	1.60E+06	4.532	0.50852
		NCFET 15nm	0.4	8747.97458	1.34E-02	0.732	2.57E+05	1.002	0.48184
	SHA256	Baseline FET 15nm	0.8	4999.69226	4.35E-02	1.749	8.45E+05	2.638	0.62211
		NCFET 15nm	0.4	5001.51088	1.13E-02	0.445	1.26E+05	0.582	0.61939
	KECCAK	Baseline FET 15nm	0.8	10093.658	0.313	3.384	1.95E+06	5.648	0.38193
		NCFET 15nm	0.4	9991.96254	7.70E-02	0.841	3.16E+05	1.233	0.40805
	Karatsuba Multiplier	Baseline FET 15nm	0.8	21642.2643	2.183	9.911	4.40E+06	16.498	3.32799
		NCFET 15nm	0.4	21537.718	0.531	2.373	7.01E+05	3.605	3.20221

Table 5.5: Comparison of Synthesis Results for Cryptographic Modules. Area is in um^2. Swiching, Internal and Total power are in mW. Leakage power is in nW. Critical Delay is in ns.

presents a qualitatively different set of trade-offs for computer architects to explore. The traditional relationship between technology area and energy efficiency (smaller CMOS gates use less energy per switch) is inverted (larger superconducting technologies can use less energy). Specifically, in this paper, we examine analytically the energy-efficiency of two leading superconducting technologies (ERSFQ and AQFP) using both technologies in the context of a hypothetical multi-core machine, and explore the potential for hybrid design (combining both technologies) to further improve efficiency over the use of either technology on its own. Considering that die area needed to implement any superconducting electronics device is large compared to CMOS, we use area as a first order constraint and devise an area-equivalent model to compare different core configurations. We show that a hybrid design has significant potential to improve energy efficiency over technology-homogeneous designs in the case that there is sufficiently large space for a number of AQFP cores. As chip area budget approaches wafer-scale, and with Amdahl fraction (parallelizable portion of the program) f=0.9, energy-efficiency improvements of 2x could be expected from hybrid approaches.

5.2.1 Introduction

As Moore's Law and traditional device scaling ends, the push for continued systems performance scaling becomes even more challenging. While architects have recently exploited chip specialization to compensate for limited device scaling [7], this too has its own limitations[8], which encourages exploring device technologies beyond traditional CMOS. Superconducting electronics (SCE) is a promising "Post-Moore" option because it allows ultra-fast switching at low energy compared to CMOS. Superconducting devices have been well-studied and there have been many proposed variations such as rapid single-flux-quantum (RSFQ) [9], energy-efficient RSFQ (ERSFQ) [10], and more recently, adiabatic quantum-flux parametron (AQFP)[11]. While significant progress have been made in advancing SCE material, device and circuit properties, there has been little progress in understanding the architectural and system-level implications of such technology.

Some large-scale RSFQ integrated circuits are rapidly developed, which includes singe-precision floating-point units [141], single-chip FFT processor [142], reconfigurable data paths, 8-bit microprocessors with memory [37, 38], 4-bit sliced ALU for 32-bit RSFQ processors.[143] Some of the processors are designed with AQFP, such as the Monolithic AQFP microprocessor. [144]

The current fabrication process[145] for basic logic three dimensional AQFP is the AIST 10 kA cm^{-2} Nb double gate process, which means that two active layers are separated by ground plane. The reasonable excitation margins for 3-D XOR gate is measured around ±16%. The fabricated chip was used for designing AQFP-based RISC-V ALU[39].

A review of an 8-Nb-layer MIT-LL fabrication process [146] for two very large scale integration SFQ digital circuit was conducted. Tolpygo et al. indicated the result for fabricating SFQ4ee ("ee" means tuned to energy-efficiency) has yielded the largest JJ counts on a single chip by their time. The statistical measurement observed that topography created by pattern wires caused less than 1% increase of the mean conductance of the junctions under a room temperature environment.

In this work, we use analytical models to explore the performance and power benefits of homogeneous SCE multicore designs as well as a hybrid heterogeneous multicore design. We identify area as a first order constraint, rather than power, and devise an area-equivalent model to compare different core configurations. We show that the hybrid design is more energy efficient than the non-hybrid designs when there are a sufficiently large number of AQFP cores. Our work serves as early guidance for design space exploration of the limitations and potential for multicore systems to be built from these superconducting technologies.

5.2.2 Superconducting Logic Families

Superconducting electronics use materials such that at least some parts of which are in superconducting state. Since superconducting electronics require to maintain at certain temperature due to their unique physics characteristics, the common temperatures for superconducting devices are the boiling point of liquid nitrogen, the boiling point of liquid helium, and the superfluid helium-4 temperature, which is below 2.17K. Although this is a crucial setting for current computers, their performance and energy-efficiency are promising for the future post-Moore research in computer architecture. For example, 8-bit AQFP adder reported a 24 k_bT energy dissipation per junction [36] .

Superconducting technology is based on the Josephson Junction (JJ), a primitive switching device. A JJ is composed of an insulating barrier that is sandwiched between two superconducting layers. In its superconducting state, despite no voltage applied across the junction, tunnelling current can pass through the junction. Once a certain current limit (I_c, critical current) is flowing through the junction, it switches to its resistive state.

RSFQ and AQFP are two leading candidate superconducting technologies. The clock frequency for RSFQ can reach 50 GHz for 8-bit processors[37]. Although RSFQ can operate at high clock frequencies, it still has a significant leakage from its bias resistors used to supply DC currents. This drastically increases the static power dissipated between 10 and 100 times the dynamic power. An energy-efficient RSFQ (ERSFQ) was proposed to address this by removing leakage at the expense of larger area by using larger bias inductors with different power distribution instead of resistors [38].

AQFP makes use of AC bias for its clock and power supply, unlike other superconducting technologies like RSFQ which are based on DC bias, allowing it to avoid DC power overhead and essentially consume less power [39]. An SFQ (Single Flux Quantum) is then stored in either left or right loop depending on the input current I_{in}. Note that, compared to RSFQ, which use JJ switching to move SFQ, information in AQFP is encoded by the location of the SFQ, which determines whether it represents logical '1' or '0'. Inverter and constant cells can be generated from this buffer cell. This set of 3 cells can then be used to build logic gates such as MAJ (majority), NOR, and AND. As a result of having the same AC signal as a power source and clock, a clocking scheme is needed to synchronize the outputs of all gates in the same clock phase. Typically, individual AQFP logic gates are connected to an AC clock signal and each one will occupy a clock phase.

Table 5.6: Energy-Delay Comparison of Superconducting Technologies

Technology	Energy (aJ)	Delay (ps)	EDP (aJ ps)
ERSFQ[147]	1400	8300	1.164E7
AQFP [148]	7.74E-4	200	0.155

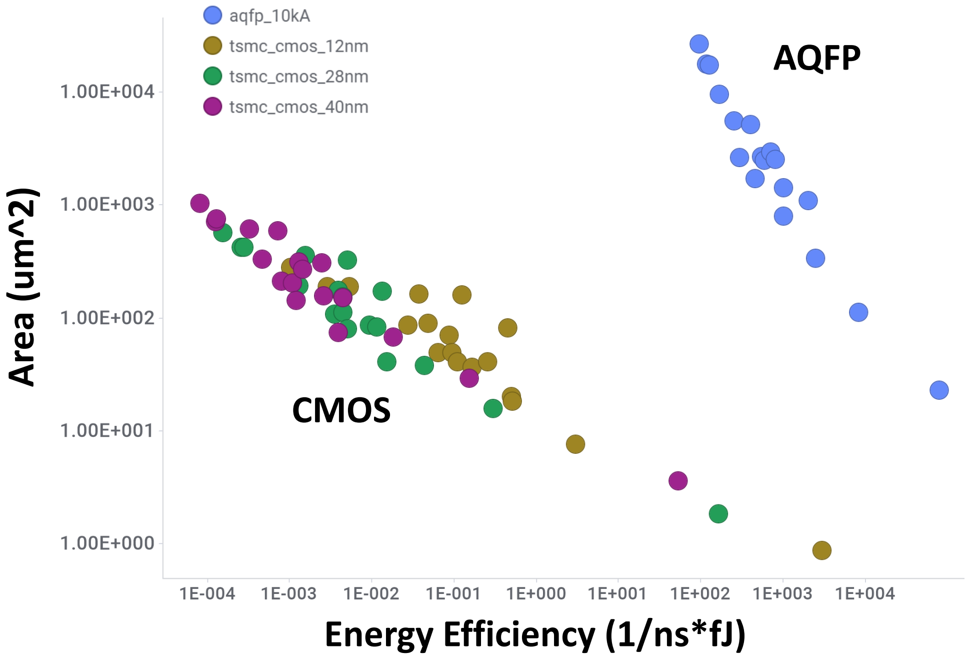


Figure 5.9: AQFP vs. CMOS in Energy-Delay Performance. AQFP has significant advantage in energy efficiency but at the cost of large area. Raw data obtained from Chen et al. [39]

5.2.3 Related Work

Hill and Marty[149] proposed an early-stage performance modeling technique for multicore processors. They complemented the Amdahl's Law software model by proposing a simple multicore hardware resources model on symmetric, asymmetric, and dynamic multicore designs. However, this model is limited to considering differences in performance between different purpose processors on chip since this model assumes all cores are equivalent. Woo and Lee [150] extended Hill and Marty's models to include power as the main consideration. Using the same P (high-performance) and c (energy efficient) core models, they proposed new variables such as idle power ratio and power gap ratio to fully model power efficiency and energy efficiency of heterogeneous multicore systems.

Ayala et al. [144] built a prototype 4-bit AQFP processor, which demonstrated the state-of-the-art AQFP architecture design. Their work integrated both adiabatic data processing and memory on a single chip, which consists of switching energy of 1.4 zJ when driven by a 4-phase 5GHz AC clock at 4.2K.

Esmaeilzadeh et al.[151] formalized the notion of "Dark Silicon" which indicates that the continuous increase in the number of cores on a chip will run into performance limits because of energy usage and heat capacity, limiting the number of transistors that can be activated simultaneously. They proposed a framework for estimating the limited speedup projections and the amount of dark silicon based from device, core and multicore scaling models.

Ishida et al. [147] proposed a design for a Neural processing unit (NPU) using RSFQ and ERSFQ technologies and showed that the simple control flow in DNN applications is a good fit for their gate-level pipelining nature. They proposed a simulation framework which starts from gate-level until NPU architecture level. They validate their model by comparisng it with a fabricated MAC unit measured in 4K temperature.

Tannu et al. [152] used open-source design tools to analyze as well as analytic models to estimate performance, power and area of RQL-based SHA256 accelerator.

Takeuchi et al. [11, 153] proposed analytical models to estimate the energy efficiency of adiabatic superconducting logic and compared with other superconducting logic families.

Yamae et al. [154] pointed out their previous model for evaluating the energy dissipation of basic AQFP logic was not applicable to complex design. They proposed a new method for evaluating the heat dissipation for complex AQFP design, which is done by subtracting the energy dissipation for peripheral buffer from the entire circuit. They performed transient analysis using Josephson circuit simulator (JSIM) on a majority gate (MAJ).

Chen et al. presented AQFP's potential as a future technology for building an extremely energy efficient computing systems due to the low energy dissipation. Their benchmark synthesis on AQFP 10kA processor for RISC-V 32-bit ALU can reach 0.043076487 fJ in EPC and 0.008615297 $fJ \cdot ns$ in EDP.

Cai et al. [155] found that AQFP is compatible with stochastic computing(SC) technique due to the two properties of AQFP: difficulty to avoid RAW hazards and true random number generation(RNG) with a single AQFP buffer. They proposed a stochastic-computing deep learning framework with AQFP and redesigned the neural network components in SC-based DNN to suit for AQFP. Their results showed AQFP based-DNN performed 6.9×10^4 times higher energy efficient compared to CMOS with 96% accuracy on the MNIST dataset.

Nagaoka et al. [156] introduced the difficulty of gate-level pipelining on complex AQFP design due to timing design. Their work showed the maximum potential of SFQ logic by demonstrating high-throughput multiplier with based on a bit-parallel, gate-level-pipelined structure. Their results showed SFQ-based multiplier can be at upt 48 GHz with 5.6 mW Power. Nagaoka et al. [156] tested the SFQ-based multiplier to perform 8×8-bit signed multiplication every clock cycle on a gate-level-pipelined structure. Their result demonstrated up to 48GHZ with 5.6 mW power consumption.

Naoki et al. [153] proposed analytical models to estimate the energy efficiency of adiabatic superconducting logic and compared with other superconducting logic families.

5.2.4 Technology and Core Models

In this work, we adapt Woo-Lee's power-aware multicore model [150] to capture the difference in power contributions between ERSFQ and AQFP-based multicore processors. We start by describing the architecturally symmetric model where all cores are similar, then proceed with an asymmetric design composed of heterogenous technologies.

Symmetric Multicore Models

Amdahl's Law states that, for a program with fraction f parallelizable ($f \in [0,1]$), and N as the number of processing units, the maximum speedup is given by:

$$Perf = \frac{1}{(1-f)+\frac{f}{N}} \tag{5.3}$$

By introducing a variable k, representing the idle state to active state core power ratio ($k \in [0,1]$), and assigning the active state power as 1, the total power of a symmetric core can be modeled. During the serial portion of the program $(1-f)$ where only one core is active and others are idle, the power consumption is $1+(N-1)k$ while in the parallel portion, where all N cores are active, power consumption is N. Thus, the average power consumption can be written as:

$$Power = \frac{1+(N-1)k(1-f)}{(1-f)+\frac{f}{N}} \tag{5.4}$$

Power efficiency ($Perf/Power$) can be computed from these two previous equations and is equal to the inverse of energy since $Perf$ is the inverse of execution time. Similarly, we can compute energy efficiency ($Perf/Energy$) from $Perf$ and $Power$ since $Energy = Power * (1/Perf)$. This represents the inverse of energy-delay-product (EDP).

To properly compare models of cores from different technologies, additional parameters are needed. If we assume that s_c ($s_c \in [0,1]$) represents the **performance ratio** of a core that is more energy-efficient compared to a high-performance core, then $Perf$ can be rewritten as:

$$Perf = \frac{s_c}{(1-f)+\frac{f}{N}} \tag{5.5}$$

Additional parameters are also needed to obtain power and energy efficiency. If we assume w_c is the **power consumption ratio** of the energy-efficient core and high-performance core ($w_c \in [0, 1]$) and k_c is the **idle state to active state core power ratio** ($k_c \in [0, 1]$) of the energy efficient core, *Power*, *Perf*/*Power*, *Perf*/*Energy* from previous equations can be derived as the following equations. The derivation of these equations is explained in detail in the Woo-Lee paper [150].

$$\boxed{\begin{aligned}
Power &= \frac{w_c + (N-1)w_c k_c(1-f)}{(1-f)+\frac{f}{N}}\\
\frac{Perf}{Power} &= \frac{s_c}{w_c + (N-1)w_c k_c(1-f)}\\
\frac{Perf}{Energy} &= \frac{s_c}{(1-f)+\frac{f}{N}} \times \frac{s_c}{w_c + (N-1)w_c k_c(1-f)}
\end{aligned}}$$

Asymmetric Multicore Model

To model a technology-heterogenous multicore, we assume a single high-performance core that operates on the serial portion and a group of $(N-1)$ energy-efficient cores operating on the parallel portion. This assumes that the high performance core is idle during the parallel portion of the program. Taking this into account, the previous equations can be extended as follows [150]:

$$\boxed{\begin{aligned}
Perf &= \frac{1}{(1-f)+\frac{f}{(N-1)s_c}}\\
Power &= \frac{(1-f)1 + (N-1)w_c k_c + \frac{f}{s_c}\frac{k}{N-1} + w_c}{(1-f)+\frac{f}{(N-1)s_c}}\\
\frac{Perf}{Power} &= \frac{1}{(1-f)+\{1+(N-1)w_c k_c\}+\frac{f}{s_c}\{\frac{k}{(N-1)+w_c}\}}\\
\frac{Perf}{Energy} &= \frac{1}{(1-f)\frac{f}{(N-1)s_c}} \times \frac{1}{(1-f)\{1+(N-1)w_c k_c\}\frac{f}{s_c}\{\frac{k}{(N-1)}+w_c\}}
\end{aligned}}$$

Asymmetric Multicore Model: Parallel Mode Active High Performance Cores

In the previous model, the lone high-performance core is assumed to be only active during the sequential portion and idle during the parallel portion. We turn our attention to a model of parallel program which can operated upon even by different types of cores. In this type of workload, we assumed that each thread has about equal work which can be handled by either a high-performance core or an energy-efficient core. This means that high-performance cores will be still be used instead of being idle, but will be underutilized compared to their sequential state mode. The number of high performance cores could be more than 1 and represented by n_p and the number of high-efficiency core is n_c. These cores are active during parallel portion of the program. Like earlier, we can derive a set of equations describing performance, power, power efficiency and energy efficiency for this type of model.

$$\boxed{\begin{aligned}
Perf &= \frac{1}{(1-f)+\frac{f}{(n_p+s_c n_c)}}\\
Power &= \frac{(1-f)(1+(n_c k_c w_c)+(n_p-1)k_p)+\frac{f(n_p k_p+w_c n_c)}{(s_c n_c+n_p)}}{(1-f)+\frac{f}{(n_p+s_c n_c)}}\\
\frac{Perf}{Power} &= \frac{1}{(1-f)(1+(n_c k_c w_c)+(n_p-1)k_p)+\frac{f(n_p k_p+w_c n_c)}{(s_c n_c+n_p)}}\\
\frac{Perf}{Energy} &= \frac{1}{(1-f)+\frac{f}{(n_p+s_c n_c)}}\times\frac{1}{(1-f)(1+(n_c k_c w_c)+n_p-1)k_p)+\frac{f(n_p k_p+w_c n_c)}{(s_c n_c+n_p)}}
\end{aligned}}$$

A summary of the parameters used in the models is listed in table 5.7.

ERSFQ and AQFP Multicores

Although the asymmetric power models were originally developed considering microarchitectural heterogeneity where a big and powerful core can handle the serial part of the computaion and smaller cores handle the parallel portion, in this work we consider

Table 5.7: Parameters used in Model Equations

Parameter	Description
f	program parallelizable portion
N	number of cores
n_p	number of high-performance (P) cores
n_c	number of energy efficient (c) cores
s_c	normalized c to P performance ratio
w_c	normalized c to P power ratio
k	normalized idle power of P
k_c	normalized idle power of c

technology heterogeneity where cores are built from different underlying device technologies. This type of technology heterogeneity was only considered for cases where emerging Post-Moore devices such as TFET[157] and NCFET[35] were coupled with CMOS. There has also been some progress on integrating these different technologies in a monolithic fashion or chiplet-based designs.

We consider ERSFQ and AQFP as our superconducting logic families for this type of integration because the fast operation of ERSFQ complements the ultra-low energy operation of AQFP. Together, they present an interesting tradeoff since the smaller ERSFQ core has better performance but the larger AQFP core has better energy efficiency. This is different from the case in CMOS where faster cores typically use larger area. Thus, we explore a system composed of a multicore chip where we use ERSFQ cores as high performance cores (but higher energy) for serial processing and pair it with AQFP cores that require less power (with lower performance) for parallel processing, considering superconducting technology area constraints.

A key parameter in the heterogeneous model is the power ratio (w_c) which is the normalized ratio of the active power of the efficiency core to the active power of the performance core. In order to get the power ratio between AQFP and ERSFQ, we first calculate the switching energy for each gate. The energy of an AQFP buffer operating at

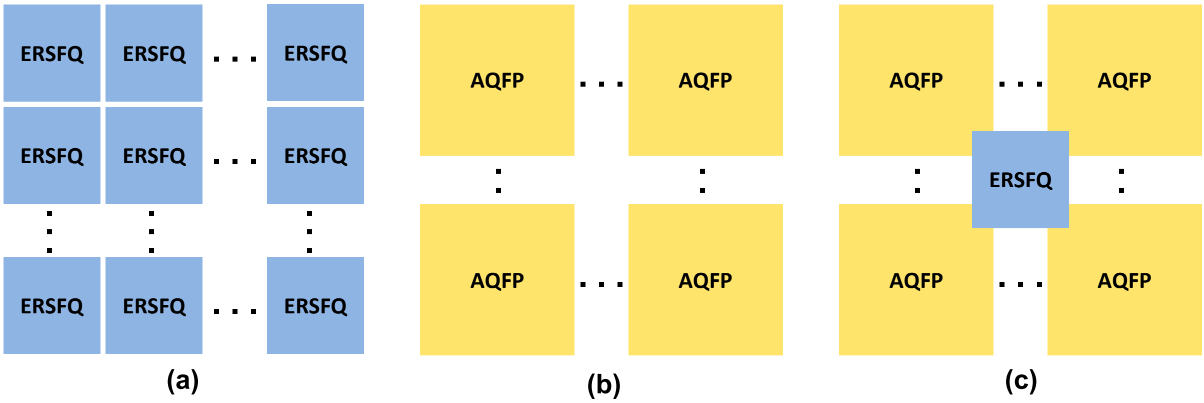


Figure 5.10: Different multicore configurations compared: (a) symmetric ERFSQ, (b) symmetric AQFP, and (c) Hybrid ERSFQ/AQFP

5 GHz is 0.774 zJ based on an 8-metal layer 100 $\mu A\mu m2$ SFQ5ee process. Assuming each gate consumes 9.97 zJ[144] and buffer overhead of 2.5[144], AQFP gate switching energy energy is estimated to be around 12 zJ. Operating at 5 GHz, the estimated power is 0.06 nW. For ERSFQ, each use a switching energy of 1.4 aJ and delay of 20 ps (50GHz), we arrive at power of 140 nW. Thus, we assume the power ratio between ERSQ and AQFP is around 2333×.

Table 5.8: Relative Ratio of Performance and Energy of ERSFQ and AQFP

Technology	Functional Unit	
	Performance	Power
ERSFQ	50 GHz	140 nW
AQFP	5∼20 GHz	0.06 nW
Gap	2.5∼10x	2333x

Area-Equivalent Model

In order to compare different configurations fairly, previous work [150] based on CMOS used power-equivalent models by normalizing to a chip power budget instead of simply comparing designs with the same core count. This is done since in CMOS, power is the main design constraint in this technology which primarily determines the effective single chip core count. However, this is not necessarily applicable in superconducting technology because of the significantly lower power consumption, even considering cooling power. Instead, in superconducting technology, area will be the main limiting constraint even before power constraints. This is because of large clock distribution networks as well as inherently larger devices and the much lower level of integration of these technologies compared to the well-developed CMOS technology, resulting in significantly larger chip areas. If we consider power as the only limiting factor, it results in hundreds of thousands of cores, each one of which is significantly larger than typical CMOS cores. Therefore, in this work, we use area as the main constraint for scaling the number of cores. In order to construct an area-equivalent model, we introduce a new parameter ac which is the **area ratio** of the performance (ERSFQ) core to the efficiency (AQFP) core as $ac = \frac{c\_area}{P\_area}$.

We can set the area budget as the number of the larger core (AQFP) $n_c = area\_budget$ and calculate the equivalent number of the smaller cores (ERSFQ) using $n_p = \frac{area\_budget}{ac}$. These core numbers can then be plugged in the the previous equations presented.

5.2.5 Evaluation and Discussion

Methodology

We used analytical models derived above to explore the performance and power benefits of SCE multicores composed of a single technology as well as evaluate a proposed hybrid design composed of multiple heterogeneous technologies. We use area as a first order constraint instead of power and devise an area-equivalent model to compare different core configurations.

We place a chip area budget parameter which determines the total number of cores for both types. The maximum chip area constraint we consider is 4.66E10 μm^2 from

Cerebras WSE-2, the largest chip in CMOS to date [158]. For ERFSQ, we use a core area of 5.94E6 μm^2 from a recent RSFQ 8-bit processor[37]. For AQFP core, we use an area of 1.6E8 μm^2 representing area from the estimated size of a state-of-the-art 4-bit AQFP processor [144] scaled to 8 bits. Thus, we can see that the ERSFQ core is at least two orders of magnitude smaller compared to the AQFP core (27x). In comparison, typical CMOS core area such as Intel Skylake 14nm is 1.19E7 μm^2[159]. The area budget is swept until 350 AQFP core area units which is a value close to the maximum number of the AQFP cores given the maximum chip area constraint.

In this work, we did not consider the overheads of interfacing between RSFQ and AQFP, but recent work [160, 161] have demonstrated circuits which make this feasible and only incur small area and power overheads. Since this paper focuses mainly on the potential of combining AQFP and ERSFQ cores, the power consumption of the uncore components were also not considered and the assumed number of uncore components will scale with the design.

We use energy-normalized performance (perf/joule) as the metric to compare the symmetric and hybrid core configurations. If energy is not a concern, symmetric configurations of high performance cores (ERFSQ) will obviously yield maximum performance.

Results and Discussion

Fig. 5.11 shows the normalized perf/joule for various configurations as a function of chip area budget for various f (parallelizable program portion) values. We also include hybrid values for power ratios (w_c) of $3x$ and $1/3x$. This represents potential error ranges for overall chip power ratio estimation from device power ratio shown in Section 5.2.4. At lower chip area (core count budget), both symmetric configurations offer better perf/joule compared to the hybrid configuration. However, with a larger chip area budget, the hybrid configuration can have better perf/joule since more cores of the larger but more

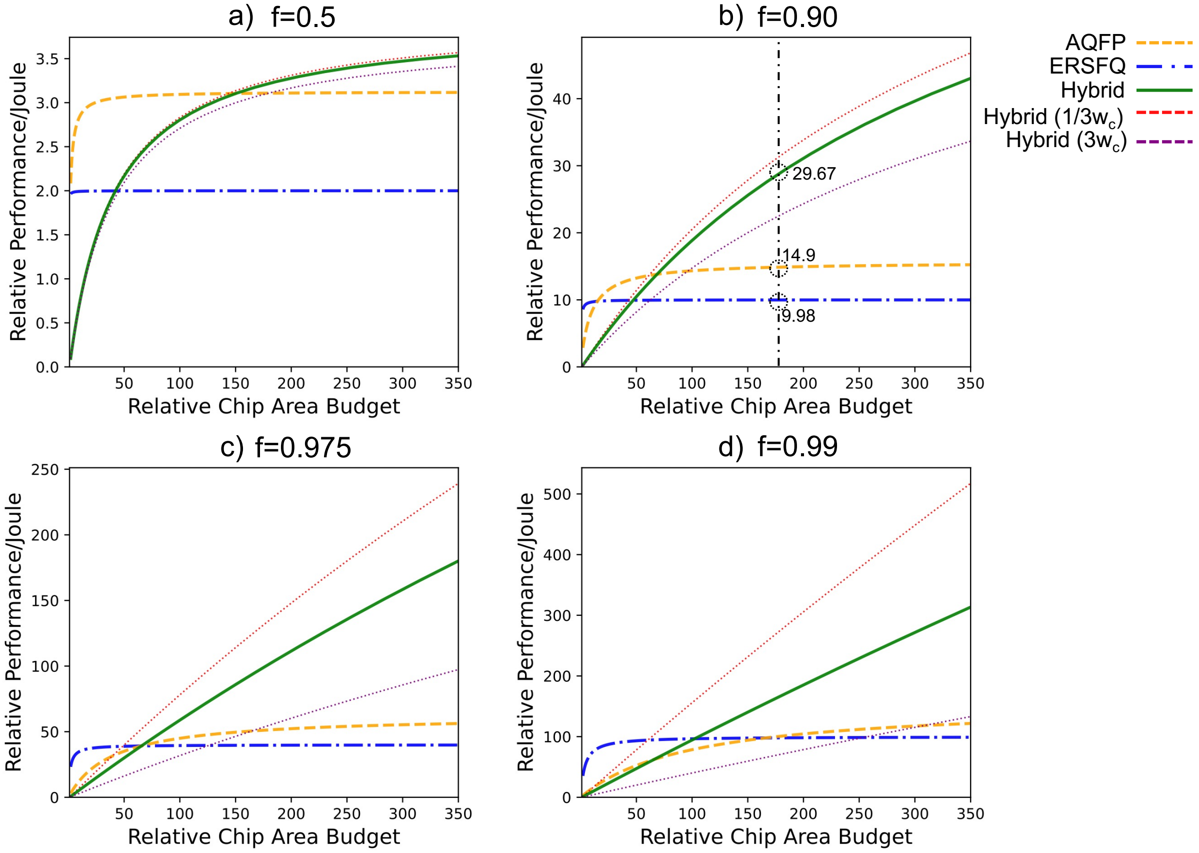


Figure 5.11: Perf/Joule as a function of chip area for various program parallelizability (0.5, 0.9, 0.975, 0.99). Chip area budget refers to number of AQFP cores.

energy-efficient AFQP can be included. In particular, we observe that this hybrid design results in better energy efficiency, if there is sufficiently large number of AQFP cores (e.g., at $f = 0.9$, more than 70 AQFP cores). As more AQFP cores are included in the hybrid design, this energy is further improved and reaches around 2x as shown in the $f = 0.9$ case in Fig. 5.11b). At low f, a symmetric AQFP is always better than symmetric ERSFQ. At high f, symmetric ERSFQ is better when the area budget is constrained. As more area is made available, eventually enough AQFP cores to meaningfully exploit parallelism can be included, making it more overall energy-efficient. Improvements to AQFP density would significantly alter this tradeoff, and is would be a natural point for future technologist to strive to reduce if they want to enable energy efficient hybrid designs. While these technologies are still under development, our results show that there is a surprising and fundamental set of trade-offs between these technology choices, distinct from traditional big/little CMOS core design considerations, and that hybrid configurations have the potential to provide the advantages of both but only as area becomes a less constrained resource.

Chapter 6

Summary of Contributions and Future Work

6.1 Summary of Contributions

In Chapter 3, we proposed an asymmetric approach to multi-party architecture with the co-location of a small physically-hardened compute element (under the control of one party) with a much larger and robust server-class system (under the control of the other). We call our proposed devices “Hardware Embassies”, a new class of devices that enable more efficient MPC by providing untrusted server co-located tamper-proof trusted hardware.

In Chapter 4, we proposed a near-data processing (NDP) architecture to accelerate privacy-preserving biomarker search. We adopt a 3D-stacked DRAM to reduce data movement and accelerate basic additive homomorphic operation for this application.

Lastly, in Chapter 5, we used presented a method for rapid evaluation of Post-Moore technology-based accelerators as well as analytical models to explore the performance and power benefits of homogeneous SCE multicore designs as well as a hybrid heterogeneous multicore design. This serves as early guidance for design space exploration of the limitations and potential for multicore systems to be built from these superconducting and other Post-Moore technologies.

6.2 Future Work

Through this series of work, we showed that some interesting sets of architecture and technologies can be used to improve some emerging datacenter cryptographic applications. A few more interesting direction for future work are as follows:

Stronger Security Assumptions

In this work, we used an honest-but-curious (semi-honest) assumption which based on traditional Garbled Circuits which might not be ideal in actual applications. A stronger assumption on the security of the system is more practical. Active adversary security is a key challenge for many security applications and protocols. As an example, a Dual Execution protocol [162] can be used to strengthen the security assumption as show in Figure 6.1. In this protocol, two independent runs of semi-honest Garbled Circuits are performed where in one run one party acts at the garbler and on the next run acts as evaluator (and the opposite for the other party). The outputs are then compared outputs at the end of the execution. The only caveat is that is a single bit (comparison result) is leaked which might be tolerable for many applications. In this protocol, more communication overhead is expected (essentially running the whole Garbled Circuit twice) which increases motivation for using Embassies.

Multiparty Computation (N>2)

Garbled Circuit is a type of MPC with only two parties (2PC). Other MPC protocols such as BGW, SPDZ, and BMR are needed when the number of parties is greater than two. However, they have practical limitations. First, as in Garbled Circuit, they

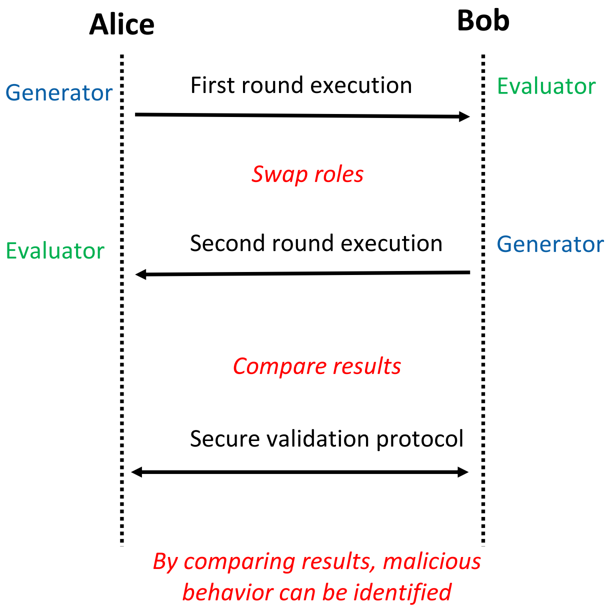


Figure 6.1: Dual Execution Flow [162]

generally require that all participating parties to be online during the protocol execution which is harder to enforce as the number of parties involved increases. Second, the number of rounds of communication in the protocol grows with the complexity of the computation and the total bandwidth also scales quadratically as the number of parties. Thus, Embassy can help address these two challenges, as depicted in Figure 6.2, since it can essentially create local proxies for the parties to execute the protocol within a datacenter.

Heterogenous Integration with other Accelerators

The inherent limitations in trusted hardware like small system resources make it difficult to scale for larger programs. Other hardware resources are available in modern datacenters such as accelerators (FPGA, GPU, ASICs) can be used for supporting secure computation. We envision a heterogenous secure datacenter architecture where parts of the secure computation can be offloaded to accelerators. A trusted hardware similar to Embassy can be used to orchestrate various trusted accelerators (widened secure base)

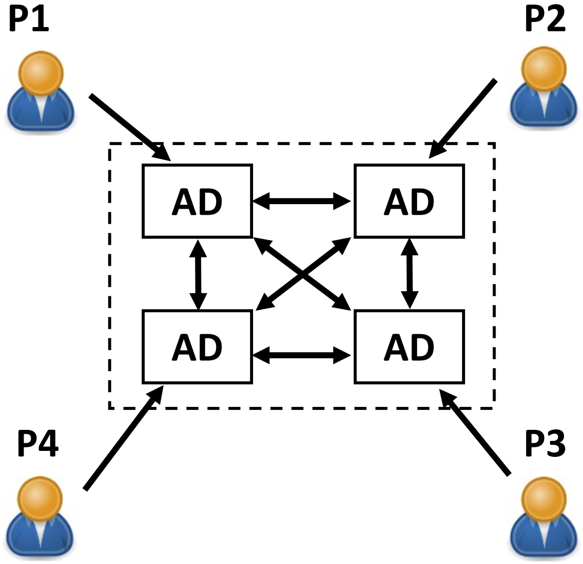


Figure 6.2: Multiparty (N>2) Configuration

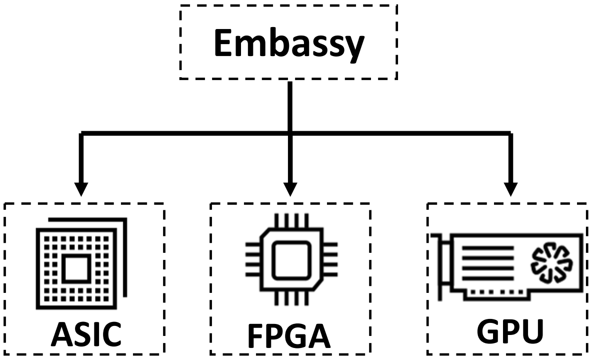


Figure 6.3: Embassy as an Secure Orchestrator for Heterogenous Accelerators

as shown in Figure 6.3.

Bibliography

[1] C. Gentry, *Computing arbitrary functions of encrypted data*, *Communications of the ACM* **53** (2010), no. 3 97, [arXiv:1111.6189].

[2] P. Martins, L. Sousa, and A. Mariano, *A Survey on Fully Homomorphic Encryption*, *ACM Computing Surveys* **50** (Dec, 2017) 1–33.

[3] L. A. Barroso, U. Holzle, P. Ranganathan, and M. Martonosi, *The Datacenter As a Computer: Designing Warehouse-Scale Machines.* Morgan and Claypool Publishers, 3rd ed., 2018.

[4] M. S. Riazi, K. Laine, B. Pelton, and W. Dai, *Heax: An architecture for computing on encrypted data*, in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '20, (New York, NY, USA), p. 1295–1309, Association for Computing Machinery, 2020.

[5] N. Samardzic, A. Feldmann, A. Krastev, S. Devadas, R. Dreslinski, C. Peikert, and D. Sanchez, *F1: A fast and programmable accelerator for fully homomorphic encryption*, in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '21, (New York, NY, USA), pp. 238–252, Association for Computing Machinery, Oct., 2021.

[6] B. Reagen, W.-S. Choi, Y. Ko, V. T. Lee, H.-H. S. Lee, G.-Y. Wei, and D. Brooks, *Cheetah: Optimizing and accelerating homomorphic encryption for private inference*, in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 26–39, 2021.

[7] W. J. Dally, Y. Turakhia, and S. Han, *Domain-specific hardware accelerators*, *Commun. ACM* **63** (June, 2020) 48–57.

[8] A. Fuchs and D. Wentzlaff, *The accelerator wall: Limits of chip specialization*, in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 1–14, Feb., 2019.

[9] K. K. Likharev and V. K. Semenov, *RSFQ logic/memory family: a new josephson-junction technology for sub-terahertz-clock-frequency digital systems*, *IEEE Trans. Appl. Supercond.* **1** (Mar., 1991) 3–28.

[10] O. A. Mukhanov, *Energy-Efficient single flux quantum technology*, *IEEE Trans. Appl. Supercond.* **21** (June, 2011) 760–769.

[11] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, *Simulation of sub-kBT bit-energy operation of adiabatic quantum-flux-parametron logic with low bit-error-rate*, *Appl. Phys. Lett.* **103** (Aug., 2013) 062602.

[12] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank, *Extending oblivious transfers efficiently*, in *Advances in Cryptology - CRYPTO 2003*, pp. 145–161, Springer Berlin Heidelberg, 2003.

[13] G. Asharov, Y. Lindell, T. Schneider, and M. Zohner, *More Efficient Oblivious Transfer and Extensions for Faster Secure Computation**, tech. rep.

[14] V. Kolesnikov and R. Kumaresan, *Improved OT Extension for Transferring Short Secrets*, tech. rep.

[15] B. Pinkas, T. Schneider, G. Segev, and M. Zohner, *Phasing: Private set intersection using permutation-based hashing*, in *24th USENIX Security Symposium (USENIX Security 15)*, (Washington, D.C.), pp. 515–530, USENIX Association, Aug., 2015.

[16] A. C.-C. Yao, *How to generate and exchange secrets*, in *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pp. 162–167, IEEE, oct, 2008.

[17] V. Kolesnikov and T. Schneider, *Improved garbled circuit: Free xor gates and applications*, in *Automata, Languages and Programming* (L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, eds.), (Berlin, Heidelberg), pp. 486–498, Springer Berlin Heidelberg, 2008.

[18] M. Naor, B. Pinkas, and R. Sumner, *Privacy preserving auctions and mechanism design*, in *Proceedings of the 1st ACM Conference on Electronic Commerce*, EC '99, (New York, NY, USA), pp. 129–139, ACM, 1999.

[19] M. Bellare, V. T. Hoang, S. Keelveedhi, and P. Rogaway, *Efficient garbling from a fixed-key blockcipher*, in *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, SP '13, (Washington, DC, USA), pp. 478–492, IEEE Computer Society, 2013.

[20] S. Zahur, M. Rosulek, and D. Evans, *Two Halves Make a Whole*, in *Advances in Cryptology - EUROCRYPT 2015* (E. Oswald and M. Fischlin, eds.), (Berlin, Heidelberg), pp. 220–250, Springer Berlin Heidelberg, 2015.

[21] E. M. Songhori, S. U. Hussain, A. R. Sadeghi, T. Schneider, and F. Koushanfar, *TinyGarble: Highly compressed and scalable sequential Garbled Circuits*, *Proceedings - IEEE Symposium on Security and Privacy* **2015-July** (2015) 411–428.

[22] O. Goldreich, S. Micali, and A. Wigderson, *How to play any mental game*, in *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, (New York, NY, USA), pp. 218–229, ACM, 1987.

[23] G. Dessouky, F. Koushanfar, A.-R. Sadeghi, T. Schneider, S. Zeitouni, and M. Zohner, *Pushing the Communication Barrier in Secure Computation using Lookup Tables*, in *Proceedings 2017 Network and Distributed System Security Symposium*, (Reston, VA), Internet Society, 2017.

[24] C. Gentry, *a Fully Homomorphic Encryption Scheme*, *PhD Thesis* (2009), no. September 1–209.

[25] J. Van Bulck, M. Minkin, O. Weisse, D. Genkin, B. Kasikci, F. Piessens, M. Silberstein, T. F. Wenisch, Y. Yarom, and R. Strackx, *Foreshadow: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-order Execution*, in *Proceedings of the 27th USENIX Conference on Security Symposium*, SEC'18, (Berkeley, CA, USA), pp. 991–1008, USENIX Association, 2018.

[26] "Hardware-Backed Heist: Extracting ECDSA Keys from Qualcomm's TrustZone." `https://www.nccgroup.trust/us/our-research/extracting-ecdsa-keys-from-qualcomms-trustzone/`.

[27] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, *Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy*, in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 201–210, JMLR.org, 2016.

[28] P. Mohassel and Y. Zhang, *SecureML: A System for Scalable Privacy-Preserving Machine Learning*, *Proceedings - IEEE Symposium on Security and Privacy* (2017) 19–38.

[29] J. Liu, M. Juuti, Y. Lu, and N. Asokan, *Oblivious Neural Network Predictions via MiniONN Transformations*, in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, (New York, NY, USA), pp. 619–631, ACM, 2017.

[30] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, *GAZELLE: A Low Latency Framework for Secure Neural Network Inference*, in *Proceedings of the 27th USENIX Conference on Security Symposium*, SEC'18, (Berkeley, CA, USA), pp. 1651–1668, USENIX Association, 2018.

[31] M. S. Riazi, M. Samragh, H. Chen, K. Laine, K. E. Lauter, and F. Koushanfar, *XONN: XNOR-based Oblivious Deep Neural Network Inference*, in *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, USENIX Association, 2019.

[32] B. Pinkas, T. Schneider, and M. Zohner, *Scalable private set intersection based on ot extension*, *ACM Trans. Priv. Secur.* **21** (Jan., 2018).

[33] V. Kolesnikov, R. Kumaresan, M. Rosulek, and N. Trieu, *Efficient batched oblivious prf with applications to private set intersection*, in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, (New York, NY, USA), p. 818–829, Association for Computing Machinery, 2016.

[34] M. M. A. Aziz, M. N. Sadat, D. Alhadidi, S. Wang, X. Jiang, C. L. Brown, and N. Mohammed, *Privacy-preserving techniques of genomic data—a survey*, *Briefings in Bioinformatics* (2017), no. April 1–9.

[35] S. Salamin, M. Rapp, A. Pathania, A. Maity, J. Henkel, T. Mitra, and H. Amrouch, *Power-Efficient heterogeneous Many-Core design with NCFET technology*, *IEEE Trans. Comput.* (2020) 1–1.

[36] N. Takeuchi, T. Yamae, C. L. Ayala, H. Suzuki, and N. Yoshikawa, *An adiabatic superconductor 8-bit adder with 24kBT energy dissipation per junction*, *Appl. Phys. Lett.* **114** (Jan., 2019) 042602.

[37] Y. Ando, R. Sato, M. Tanaka, K. Takagi, N. Takagi, and A. Fujimaki, *Design and demonstration of an 8-bit Bit-Serial RSFQ microprocessor: CORE e4*, *IEEE Trans. Appl. Supercond.* **26** (Aug., 2016) 1–5.

[38] R. Sato, Y. Hatanaka, Y. Ando, M. Tanaka, A. Fujimaki, K. Takagi, and N. Takagi, *High-Speed operation of Random-Access-Memory-Embedded microprocessor with minimal instruction set architecture based on rapid Single-Flux-Quantum logic*, *IEEE Trans. Appl. Supercond.* **27** (June, 2017) 1–5.

[39] O. Chen, R. Cai, Y. Wang, F. Ke, T. Yamae, R. Saito, N. Takeuchi, and N. Yoshikawa, *Adiabatic Quantum-Flux-Parametron: Towards building extremely Energy-Efficient circuits and systems*, *Sci. Rep.* **9** (July, 2019) 10514.

[40] J. M. McCune, B. J. Parno, A. Perrig, M. K. Reiter, and H. Isozaki, *Flicker: An execution infrastructure for tcb minimization*, in *Proceedings of the 3rd ACM SIGOPS/EuroSys European Conference on Computer Systems 2008*, Eurosys '08, (New York, NY, USA), pp. 315–328, ACM, 2008.

[41] V. Costan and S. Devadas, *Intel sgx explained*, *IACR Cryptology ePrint Archive* **2016** (2016) 86.

[42] Y. Lindell, *Secure multiparty computation*, *Commun. ACM* **64** (Dec., 2020) 86–96.

[43] A. Herzberg and H. Shulman, *Oblivious and fair server-aided two-party computation*, in *2012 Seventh International Conference on Availability, Reliability and Security*, pp. 75–84, 2012.

[44] B. D. Rouhani, S. U. Hussain, K. Lauter, and F. Koushanfar, *ReDCrypt: Real-Time Privacy-Preserving Deep Learning Inference in Clouds Using FPGAs*, *ACM Transactions on Reconfigurable Technology and Systems* **11** (dec, 2018) 1–21.

[45] K. A. Jagadeesh, D. J. Wu, J. A. Birgmeier, D. Boneh, and G. Bejerano, *Deriving genomic diagnoses without revealing patient genomes.*, *Science (New York, N.Y.)* **357** (aug, 2017) 692–695.

[46] B. D. Rouhani, M. S. Riazi, and F. Koushanfar, *Deepsecure: Scalable Provably-Secure Deep Learning*, in *Proceedings of the 55th Annual Design Automation Conference on - DAC '18*, (New York, New York, USA), pp. 1–6, ACM Press, 2018.

[47] "Pushing the Communication Barrier in 2PC using Lookup Tables." `https://www.ndss-symposium.org/wp-content/uploads/2017/09/ndss2017-04B-3-zohner_slides.pdf`.

[48] "ITRS: System Integration." `https://www.semiconductors.org/wp-content/uploads/2018/06/1_2015-ITRS-20_System-Integration.pdf`.

[49] NIST, *Fips pub 140-2: Security requirements for cryptographic modules*, 2001.

[50] NIST, *Fips pub 140-3: Security requirements for cryptographic modules*, 2019.

[51] N. Rangarajan, S. Patnaik, J. Knechtel, S. Rakheja, and O. Sinanoglu, *Tamper-proof hardware from emerging technologies*, in *The Next Era in Hardware Security*, pp. 195–209. Springer, 2021.

[52] M. T. Rahman, Q. Shi, S. Tajik, H. Shen, D. L. Woodard, M. Tehranipoor, and N. Asadizanjani, *Physical inspection attacks: New frontier in hardware security*, in *2018 IEEE 3rd International Verification and Security Workshop (IVSW)*, pp. 93–102, 2018.

[53] V. Immler, J. Obermaier, K. Ng, F. Ke, J. Lee, Y. Lim, W. Oh, K. Wee, and G. Sigl, *Secure physical enclosures from covers with tamper-resistance*, *IACR Transactions on Cryptographic Hardware and Embedded Systems* **2019** (Nov., 2018) 51–96.

[54] “Intel Unveils the Intel Neural Compute Stick 2 at Intel AI Devcon Beijing for Building Smarter AI Edge Devices.” `https://newsroom.intel.com/news/intel-unveils-intel-neural-compute-stick-2/`.

[55] “Google Edge TPU.” `https://cloud.google.com/edge-tpu/`.

[56] “Intel Compute Stick.” `https://www.intel.com/content/www/us/en/products/boards-kits/compute-stick.html`.

[57] A. Guleria, J. Lakshmi, and C. Padala, *Quadd: Quantifying accelerator disaggregated datacenter efficiency*, in *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, pp. 349–357, 2019.

[58] P. Kocher, J. Horn, A. Fogh, , D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher, M. Schwarz, and Y. Yarom, *Spectre attacks: Exploiting speculative execution*, in *40th IEEE Symposium on Security and Privacy (S&P'19)*, 2019.

[59] M. Lipp, M. Schwarz, D. Gruss, T. Prescher, W. Haas, A. Fogh, J. Horn, S. Mangard, P. Kocher, D. Genkin, Y. Yarom, and M. Hamburg, *Meltdown: Reading kernel memory from user space*, in *27th USENIX Security Symposium (USENIX Security 18)*, 2018.

[60] M. Schwarz, M. Lipp, D. Moghimi, J. Van Bulck, J. Stecklina, T. Prescher, and D. Gruss, *Zombieload: Cross-privilege-boundary data sampling*, in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, (New York, NY, USA), pp. 753–768, Association for Computing Machinery, 2019.

[61] A. Caulfield, P. Costa, and M. Ghobadi, *Beyond smartnics: Towards a fully programmable cloud: Invited paper*, in *2018 IEEE 19th International Conference on High Performance Switching and Routing (HPSR)*, pp. 1–6, 2018.

[62] lowRISC C.I.C., “OpenTitan.” `https://opentitan.org/`.

[63] T. Schneider and M. Zohner, *GMW vs. Yao? Efficient secure two-party computation with low depth circuits*, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7859 LNCS, pp. 275–292, Springer, Berlin, Heidelberg, 2013.

[64] S. U. Hussain and F. Koushanfar, *FASE: FPGA acceleration of secure function evaluation*, in *2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pp. 280–288, Apr., 2019.

[65] “Equinix Metal.” `https://metal.equinix.com/`.

[66] "Azulle Access3." https://azulletech.com/product/access3/.

[67] "chiraag/gazelle_mpc." https://github.com/chiraag/gazelle_mpc. (Accessed on 06/09/2022).

[68] "encryptogroup/aby: Aby - a framework for efficient mixed-protocol secure two-party computation." https://github.com/encryptogroup/ABY. (Accessed on 06/09/2022).

[69] "encryptogroup/psi: Implementations of private set intersection protocols." https://github.com/encryptogroup/PSI. (Accessed on 06/09/2022).

[70] R. Zhu, D. Cassel, A. Sabry, and Y. Huang, *NANOPI: Extreme-Scale Actively-Secure Multi-Party Computation*, in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security - CCS '18*, (New York, New York, USA), pp. 862–879, ACM Press, 2018.

[71] "Verizon IP Latency Statistics." https://www.verizon.com/business/terms/latency/.

[72] "AWS Direct Connect." https://aws.amazon.com/directconnect/.

[73] M. Kim, Y. . C. Song, and J.H, *Secure searching of biomarkers through hybrid homomorphic encryption scheme*, in *BMC Med Genomics*, 2017.

[74] "Cost of Click." https://energyzarr.typepad.com/energyzarrnationalcom/2008/08/the-true-cost-o.html.

[75] P. Ruiu, C. Fiandrino, P. Giaccone, A. Bianco, D. Kliazovich, and P. Bouvry, *On the energy-proportionality of data center networks*, *IEEE Transactions on Sustainable Computing* **2** (2017), no. 2 197–210.

[76] "Aceslabucsd/tinygarblecircuitsynthesis: Circuit synthesis for yao's garbled circuit by tinygarble." https://github.com/ACESLabUCSD/TinyGarbleCircuitSynthesis. (Accessed on 06/09/2022).

[77] P. X. Gao, A. Narayan, S. Karandikar, J. Carreira, S. Han, R. Agarwal, S. Ratnasamy, and S. Shenker, *Network Requirements for Resource Disaggregation*, in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, (Berkeley, CA, USA), pp. 249–264, USENIX Association, 2016.

[78] "Reinventing our data center network with F16, Minipack - Facebook Code." https://code.fb.com/data-center-engineering/f16-minipack/.

[79] S. Yeoman, *How secure are bare metal servers?*, *Network Security* **2019** (2019), no. 2 16 – 17.

[80] F. Tramer and D. Boneh, *Slalom: Fast, verifiable and private execution of neural networks in trusted hardware*, in *International Conference on Learning Representations*, 2019.

[81] O. Ohrimenko, F. Schuster, C. Fournet, A. Mehta, S. Nowozin, K. Vaswani, and M. Costa, *Oblivious multi-party machine learning on trusted processors*, in *Proceedings of the 25th USENIX Conference on Security Symposium*, SEC'16, (Berkeley, CA, USA), pp. 619–636, USENIX Association, 2016.

[82] K. Grover, S. Tople, S. Shinde, R. Bhagwan, and R. Ramjee, *Privado: Practical and secure dnn inference with enclaves*, 2019.

[83] N. Hynes, R. Cheng, and D. Song, *Efficient Deep Learning on Multi-Source Private Data*, *arXiv e-prints* (July, 2018) arXiv:1807.06689, [arXiv:1807.0668].

[84] S. Chakrabarti, M. Hoekstra, D. Kuvaiskii, and M. Vij, *Scaling intel® software guard extensions applications with intel® sgx card*, in *Proceedings of the 8th International Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '19, (New York, NY, USA), Association for Computing Machinery, 2019.

[85] R. Bahmani, M. Barbosa, F. Brasser, B. Portela, A.-R. Sadeghi, G. Scerri, and B. Warinschi, *Secure Multiparty Computation from SGX*, in *Financial Cryptography and Data Security* (A. Kiayias, ed.), (Cham), pp. 477–497, Springer International Publishing, 2017.

[86] V. A. Sartakov, S. Brenner, S. Ben Mokhtar, S. Bouchenak, G. Thomas, and R. Kapitza, *Eactors: Fast and flexible trusted computing using sgx*, in *Proceedings of the 19th International Middleware Conference*, Middleware '18, (New York, NY, USA), pp. 187–200, ACM, 2018.

[87] D. Demmler, T. Schneider, M. Zohner, and T. Universit, *Ad-Hoc Secure Two-Party Computation on Mobile Devices using Hardware Tokens*, *USENIX Security* (2014) 893–908.

[88] S. Bugiel, S. Nürnberger, A.-R. Sadeghi, and T. Schneider, *Twin clouds: Secure cloud computing with low latency*, in *Proceedings of the 12th IFIP TC 6/TC 11 International Conference on Communications and Multimedia Security*, CMS'11, (Berlin, Heidelberg), p. 32–44, Springer-Verlag, 2011.

[89] K. Eguro and R. Venkatesan, *Fpgas for trusted cloud computing*, in *22nd International Conference on Field Programmable Logic and Applications (FPL)*, pp. 63–70, Aug, 2012.

[90] T. Hunt, Z. Jia, V. Miller, A. Szekely, Y. Hu, C. J. Rossbach, and E. Witchel, *Telekine: Secure computing with cloud gpus*, in *17th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 20)*, pp. 817–833, 2020.

[91] J. Zhu, R. Hou, X. Wang, W. Wang, J. Cao, B. Zhao, Z. Wang, Y. Zhang, J. Ying, L. Zhang, and D. Meng, *Enabling rack-scale confidential computing using heterogeneous trusted execution environment*, 2020.

[92] F. Kerschbaum, T. Schneider, and A. Schröpfer, *Automatic protocol selection in secure two-party computations*, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8479 LNCS, pp. 566–584, Springer, Cham, 2014.

[93] E. Pattuk, M. Kantarcioglu, H. Ulusoy, and B. Malin, *CheapSMC: A framework to minimize secure multiparty computation cost in the cloud*, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9766, pp. 285–294, Springer, Cham, 2016.

[94] M. von Maltitz and G. Carle, *A Performance and Resource Consumption Assessment of Secret Sharing Based Secure Multiparty Computation*, in *Data Privacy Management, Cryptocurrencies and Blockchain Technology* (J. Garcia-Alfaro, J. Herrera-Joancomartí, G. Livraga, and R. Rios, eds.), (Cham), pp. 357–372, Springer International Publishing, 2018.

[95] M. S. Riazi, C. Weinert, O. Tkachenko, E. M. Songhori, T. Schneider, and F. Koushanfar, *Chameleon: A Hybrid Secure Computation Framework for Machine Learning Applications*, in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security - ASIACCS '18*, (New York, New York, USA), pp. 707–721, ACM Press, 2018.

[96] D. Demmler, K. Hamacher, T. Schneider, and S. Stammler, *Privacy-preserving whole-genome variant queries*, in *Cryptology and Network Security* (S. Capkun and S. S. M. Chow, eds.), (Cham), pp. 71–92, Springer International Publishing, 2018.

[97] T. Schneider and O. Tkachenko, *EPISODE: Efficient Privacy-PreservIng Similar Sequence Queries on Outsourced Genomic DatabasEs*, in *Proceedings of the 2019 on Asia Conference on Computer and Communications Security - ASIACCS '19*, 2019.

[98] M. S. Riazi, K. Laine, B. Pelton, and W. Dai, *Heax: An architecture for computing on encrypted data*, in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '20, (New York, NY, USA), p. 1295–1309, Association for Computing Machinery, 2020.

[99] B. Reagen, W. Choi, Y. Ko, V. Lee, G.-Y. Wei, H.-H. S. Lee, and D. Brooks, *Cheetah: Optimizing and accelerating homomorphic encryption for private inference*, in *2021 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, IEEE, 2021.

[100] P. Mishra, R. Lehmkuhl, A. Srinivasan, W. Zheng, and R. A. Popa, *Delphi: A cryptographic inference service for neural networks*, in *29th USENIX Security Symposium (USENIX Security 20)*, pp. 2505–2522, USENIX Association, Aug., 2020.

[101] "How the Pixel 2's security module delivers enterprise-grade security." `https://www.blog.google/products/android-enterprise/how-pixel-2s-security-module-delivers-enterprise-grade-security/`.

[102] "New Arm IP Helps Protect IoT Devices from Increasingly Prevalent – Arm." `https://www.arm.com/company/news/2018/05/new-arm-ip-helps-protect-iot-devices-from-increasingly-prevalent-physical-threats`.

[103] E. M. Songhori, T. Schneider, S. Zeitouni, A. Sadeghi, G. Dessouky, and F. Koushanfar, *Garbledcpu: A mips processor for secure computation in hardware*, in *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6, June, 2016.

[104] S. U. Hussain, B. D. Rouhani, M. Ghasemzadeh, and F. Koushanfar, *MAXelerator: FPGA Accelerator for Privacy Preserving Multiply-Accumulate (MAC) on Cloud Servers*, in *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, pp. 1–6, IEEE, jun, 2018.

[105] E. M. Songhori, M. S. Riazi, S. U. Hussain, A.-R. Sadeghi, and F. Koushanfar, *Arm2gc: Succinct garbled processor for secure computation*, in *Proceedings of the 56th Annual Design Automation Conference 2019*, DAC '19, (New York, NY, USA), Association for Computing Machinery, 2019.

[106] J.-P. Hubaux, S. Katzenbeisser, and B. Malin, *Genomic Data Privacy and Security: Where We Stand and Where We Are Heading*, *IEEE Security & Privacy* **15** (2017), no. 5 10–12.

[107] K. Lauter, A. López-Alt, and M. Naehrig, *Private Computation on Encrypted Genomic Data*, in *Progress in Cryptology - LATINCRYPT 2014*, vol. 8895, pp. 3–27, Springer International Publishing, 2015.

[108] A. Khedr and G. Gulak, *SecureMed: Secure Medical Computation using GPU-Accelerated Homomorphic Encryption Scheme*, *IEEE Journal of Biomedical and Health Informatics* (2017) 1–1.

[109] G. S. Çetin, H. Chen, K. Laine, K. Lauter, P. Rindal, and Y. Xia, *Private queries on encrypted genomic data*, *BMC Medical Genomics* **10** (jul, 2017) 45.

[110] M. Kim, Y. Song, and J. H. Cheon, *Secure searching of biomarkers through hybrid homomorphic encryption scheme*, *BMC Medical Genomics* **10** (2017), no. Suppl 2.

[111] J. S. Sousa, C. Lefebvre, Z. Huang, J. L. Raisaro, C. Aguilar-Melchor, M.-O. Killijian, and J.-P. Hubaux, *Efficient and secure outsourcing of genomic data storage*, *BMC Medical Genomics* **10** (jul, 2017) 46.

[112] A. Khedr, G. Gulak, and V. Vaikuntanathan, *SHIELD: Scalable Homomorphic Implementation of Encrypted Data-Classifiers*, *IEEE Transactions on Computers* **65** (2016), no. 9 2848–2858.

[113] S. Angel, H. Chen, K. Laine, and S. Setty, *Pir with compressed queries and amortized query processing*, in *IEEE SP*, pp. 962–979, May, 2018.

[114] S. Bian, M. Hiromoto, and T. Sato, *SCAM: Secured content addressable memory based on homomorphic encryption*, in *DATE*, pp. 984–989, 2017.

[115] C. Gentry, *Fully homomorphic encryption using ideal lattices*, in *STOC*, pp. 169–178, 2009.

[116] W. Wang, X. Huang, N. Emmart, and C. Weems, *Vlsi design of a large-number multiplier for fully homomorphic encryption*, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **22** (Sept, 2014) 1879–1887.

[117] L. Ducas and D. Micciancio, *FHEW: Bootstrapping homomorphic encryption in less than a second*, in *Advances in Cryptology – EUROCRYPT 2015*, vol. 9056, pp. 617–640, 2015. arXiv:1410.3918.

[118] Hybrid Memory Cube Consortium, *Hybrid Memory Cube Specification 2.1*, 2014.

[119] S. Williams, A. Waterman, and D. Patterson, *Roofline: An insightful visual performance model for multicore architectures*, *Commun. ACM* **52** (Apr., 2009) 65–76.

[120] P. Zicari and S. Perri, *A fast carry chain adder for virtex-5 fpgas*, in *IEEE Mediterranean Electrotechnical Conference*, pp. 304–308, April, 2010.

[121] R. Hadidi, B. Asgari, B. A. Mudassar, S. Mukhopadhyay, S. Yalamanchili, and H. Kim, *Demystifying the characteristics of 3d-stacked memories: A case study for hybrid memory cube*, in *IEEE IISWC*, pp. 66–75, Oct, 2017.

[122] S. H. Pugsley, J. Jestes, H. Zhang, R. Balasubramonian, V. Srinivasan, A. Buyuktosunoglu, A. Davis, and F. Li, *Ndc: Analyzing the impact of 3d-stacked memory+logic devices on mapreduce workloads*, in *ISPASS*, pp. 190–200, March, 2014.

[123] J. Jeddeloh and B. Keeth, *Hybrid memory cube new dram architecture increases density and performance*, in *VLSI*, pp. 87–88, June, 2012.

[124] S. F. Yitbarek, T. Yang, R. Das, and T. Austin, *Exploring specialized near-memory processing for data intensive operations*, *DATE* (2016) 1449–1452.

[125] M. Drumond, A. Daglis, N. Mirzadeh, D. Ustiugov, J. Picorel, B. Falsafi, B. Grot, D. Pnevmatikatos, M. Drumond, A. Daglis, N. Mirzadeh, D. Ustiugov, J. Picorel, B. Falsafi, B. Grot, and D. Pnevmatikatos, *The Mondrian Data Engine*, in *ISCA*, pp. 639–651, 2017.

[126] M. A. Z. Alves, M. Diener, P. C. Santos, and L. Carro, *Large Vector Extensions Inside the HMC*, *Design, Automation, and Test in Europe (DATE)* (2016) 1249–1254.

[127] J. S. Kim, D. Senol Cali, H. Xin, D. Lee, S. Ghose, M. Alser, H. Hassan, O. Ergin, C. Alkan, and O. Mutlu, *GRIM-Filter: Fast seed location filtering in DNA read mapping using processing-in-memory technologies*, *BMC Genomics* **19** (May, 2018) 89.

[128] P. Martins and L. Sousa, *HPC on the Intel Xeon Phi: Homomorphic Word Searching*, in *High Performance Computing for Computational Science – VECPAR 2016*, pp. 75–88, 2017.

[129] M. Sadegh Riazi, M. Samragh, and F. Koushanfar, *CAMsure: Secure Content-Addressable Memory for Approximate Search*, *ACM Trans. Embed. Comput. Syst. Article* **16** (2017), no. 20 1–20.

[130] Y. S. Shao, B. Reagen, G. Wei, and D. Brooks, *Aladdin: A pre-RTL, power-performance accelerator simulator enabling large design space exploration of customized architectures*, in *2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, pp. 97–108, June, 2014.

[131] Y. S. Shao, S. L. Xi, V. Srinivasan, G. Wei, and D. Brooks, *Co-designing accelerators and SoC interfaces using gem5-aladdin*, in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 1–12, Oct., 2016.

[132] A. Srinivasan, G. D. Huber, and D. P. LaPotin, *Accurate area and delay estimation from RTL descriptions*, *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **6** (Mar., 1998) 168–172.

[133] S. K. Samal, S. Khandelwal, A. I. Khan, S. Salahuddin, C. Hu, and S. K. Lim, *Full chip power benefits with negative capacitance FETs*, in *2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 1–6, July, 2017.

[134] "Gurobi Solver." `https://www.gurobi.com/`.

[135] "NIST PQC 3rd Round Results." `https://www.nist.gov/news-events/news/2022/07/pqc-standardization-process-announcing-four-candidates-be-standardized-plus`.

[136] "CRYSTALS-Kyber." `https://pq-crystals.org/kyber/`.

[137] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "Algorithm specifications and supporting documentation." `https://pq-crystals.org/kyber/data/kyber-specification-round3-20210131.pdf`. Accessed: 2022-7-31.

[138] "PQClean." `https://github.com/PQClean/PQClean`.

[139] "SABER." `https://github.com/KULeuven-COSIC/SABER`.

[140] H. Nejatollahi, F. Valencia, S. Banik, F. Regazzoni, R. Cammarota, and N. Dutt, *Synthesis of flexible accelerators for early adoption of Ring-LWE post-quantum cryptography*, *ACM Trans. Embed. Comput. Syst.* **19** (Mar., 2020) 1–17.

[141] X. Peng, Q. Xu, T. Kato, Y. Yamanashi, N. Yoshikawa, A. Fujimaki, N. Takagi, K. Takagi, and M. Hidaka, *High-Speed demonstration of Bit-Serial Floating-Point adders and multipliers using Single-Flux-Quantum circuits*, *IEEE Trans. Appl. Supercond.* **25** (June, 2015) 1–6.

[142] T. Ono, H. Suzuki, Y. Yamanashi, and N. Yoshikawa, *Design and implementation of an SFQ-Based Single-Chip FFT processor*, *IEEE Trans. Appl. Supercond.* **27** (June, 2017) 1–5.

[143] G. Tang, K. Takata, M. Tanaka, A. Fujimaki, K. Takagi, and N. Takagi, *4-bit Bit-Slice arithmetic logic unit for 32-bit RSFQ microprocessors*, *IEEE Trans. Appl. Supercond.* **26** (Jan., 2016) 1–6.

[144] C. L. Ayala, T. Tanaka, R. Saito, M. Nozoe, N. Takeuchi, and N. Yoshikawa, *MANA: A monolithic adiabatic integration architecture microprocessor using 1.4zj/op superconductor josephson junction devices*, in *2020 IEEE Symposium on VLSI Circuits*, pp. 1–2, June, 2020.

[145] T. Ando, S. Nagasawa, N. Takeuchi, N. Tsuji, F. China, M. Hidaka, Y. Yamanashi, and N. Yoshikawa, *Three-dimensional adiabatic quantum-flux-parametron fabricated using a double-active-layered niobium process*, *Supercond. Sci. Technol.* **30** (June, 2017) 075003.

[146] S. K. Tolpygo, V. Bolkhovsky, T. J. Weir, A. Wynn, D. E. Oates, L. M. Johnson, and M. A. Gouker, *Advanced fabrication processes for superconducting very Large-Scale integrated circuits*, *IEEE Trans. Appl. Supercond.* **26** (Apr., 2016) 1–10.

[147] K. Ishida, I. Byun, I. Nagaoka, K. Fukumitsu, M. Tanaka, S. Kawakami, T. Tanimoto, T. Ono, J. Kim, and K. Inoue, *SuperNPU: An extremely fast neural processing unit using superconducting logic devices*, in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 58–72, Oct., 2020.

[148] Y. He, C. L. Ayala, N. Takeuchi, T. Yamae, Y. Hironaka, A. Sahu, V. Gupta, A. Talalaevskii, D. Gupta, and N. Yoshikawa, *A compact AQFP logic cell design using an 8-metal layer superconductor process*, *Supercond. Sci. Technol.* **33** (Feb., 2020) 035010.

[149] M. D. Hill and M. R. Marty, *Amdahl's law in the multicore era*, *Computer* **41** (July, 2008) 33–38.

[150] D. H. Woo and H. S. Lee, *Extending amdahl's law for Energy-Efficient computing in the Many-Core era*, *Computer* **41** (Dec., 2008) 24–31.

[151] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, *Dark silicon and the end of multicore scaling*, in *Proceedings of the 38th annual international symposium on Computer architecture*, ISCA '11, (New York, NY, USA), pp. 365–376, Association for Computing Machinery, June, 2011.

[152] S. S. Tannu, P. Das, M. L. Lewis, R. Krick, D. M. Carmean, and M. K. Qureshi, *A case for superconducting accelerators*, arXiv:1902.0464.

[153] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, *Energy efficiency of adiabatic superconductor logic*, *Supercond. Sci. Technol.* **28** (Nov., 2014) 015003.

[154] T. Yamae, N. Takeuchi, and N. Yoshikawa, *Systematic method to evaluate energy dissipation in adiabatic quantum-flux-parametron logic*, *J. Appl. Phys.* **126** (Nov., 2019) 173903.

[155] R. Cai, A. Ren, O. Chen, N. Liu, C. Ding, X. Qian, J. Han, W. Luo, N. Yoshikawa, and Y. Wang, *A stochastic-computing based deep learning framework using adiabatic quantum-flux-parametron superconducting technology*,

in *Proceedings of the 46th International Symposium on Computer Architecture*, ISCA '19, (New York, NY, USA), pp. 567–578, Association for Computing Machinery, June, 2019.

[156] I. Nagaoka, M. Tanaka, K. Inoue, and A. Fujimaki, *29.3 a 48GHz 5.6mw Gate-Level-Pipelined multiplier using Single-Flux quantum logic*, in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, pp. 460–462, Feb., 2019.

[157] B. Gopireddy, D. Skarlatos, W. Zhu, and J. Torrellas, *HetCore: TFET-CMOS Hetero-Device architecture for CPUs and GPUs*, in *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, pp. 802–815, June, 2018.

[158] "Cerebras WSE-2." `https://cerebras.net/chip/`.

[159] "Intel Skylake Comparison." `https://www.anandtech.com/show/11859/the-anandtech-coffee-lake-review-8700k-and-8400-initial-numbers/`.

[160] Y. Yamazaki, N. Takeuchi, and N. Yoshikawa, *A compact interface between adiabatic Quantum-Flux-Parametron and rapid Single-Flux-Quantum circuits*, *IEEE Trans. Appl. Supercond.* **31** (Aug., 2021) 1–5.

[161] F. China, N. Tsuji, T. Narama, N. Takeuchi, T. Ortlepp, Y. Yamanashi, and N. Yoshikawa, *Demonstration of signal transmission between adiabatic Quantum-Flux-Parametrons and rapid Single-Flux-Quantum circuits using superconductive microstrip lines*, *IEEE Trans. Appl. Supercond.* **27** (June, 2017) 1–5.

[162] Y. Huang, J. Katz, and D. Evans, *Quid-pro-quo-tocols: Strengthening semi-honest protocols with dual execution*, in *2012 IEEE Symposium on Security and Privacy*, pp. 272–284, 2012.