MODEL AND EVALUATION OF A SUPERCONDUCTING-LOGIC BASED HYBRID CPU-ACCELERATOR SYSTEM

Meenatchi Jagasivamani
Christine Fong
Kenneth Goodnow
Robert Voigt

Northrop Grumman Corporation
Mission Systems Division
1580B W Nursery Rd
Linthicum, MD, USA
{meenatchi.jagasivamani, christine.fong, kenneth.goodnow, robert.voigt}@ngc.com

ABSTRACT

Superconducting logic is one of the most promising emerging technologies to deliver "Beyond-Moore's" power-efficient capabilities for high performance computing. While the technology has progressed from nascent development to design, fabrication and test, performance at scale for a system has not been evaluated. In this paper, we present a system model of a hybrid CPU-Accelerator system using superconducting Reciprocal Quantum Logic (RQL) and evaluating against a conventional NVIDIA V100 system using the SST-GPGPUSim simulation framework. Comparisons on the execution time, system power, power-efficiency, and power-delay product are presented. The results show an overall system power efficiency gain of a 3.6x for a liquid helium-based cooling, 10x more throughput, and 100x less raw system power on a FFT kernel.

Keywords: superconducting, accelerator, RQL, SST, GPGPUSim.

1 INTRODUCTION

GPU-accelerated computing continues to play a significant role in advancing parallel multi-threaded performance by providing a new way to achieve performance improvement (ITRS 2021) and through applying domain-specific solutions. At the same time, the power delivery and power dissipation limits have driven the need to seek power-efficient solutions, even for high-performance computing applications. To that end, superconducting logic is an emerging technology that is being applied to deliver computing technology at a fraction of CMOS power cost. One type of superconducting logic makes use Reciprocal Quantum Logic (RQL) to realize digital circuits that operate at liquid Helium (He) temperature of 4K (Lee 2021). RQL logic is composed of Josephson Junctions (JJ) with reciprocal pairs of Single Flux Quanta (SFQ) pulses, and uses quantum effects to provide fast switching, zero energy cost, and low power noise.

Prior investigations in this superconducting space have been limited to small-scale processing units for feasibility demonstration of the varying functions, including processor and memory units (Manheimer 2015). In this work, we model and evaluate a larger-scale system delivering TFLOPs of computing capability and assess the impact on performance metrics of interest based on system simulations.

The rest of the paper is organized as follows. In section 2, we present a background of the system architecture that we will be modeling for this work. In section 3 and 4, we describe the simulation framework that was used to assess the system model, including model validation. In section 5, we present

the simulation results. In section 6, we report on the analysis and discussion of the results including investigations for future work. In section 7, we summarize related work, and we conclude in section 8.

2 SYSTEM ARCHITECTURE

Our system is comprised of superconducting RQL-accelerators (XPU) operating at 4K, which are driven by a host-CPU residing at a warmer temperature (233K). Figure 1 shows the overall system architecture which spans multiple temperature regions. In the cold-space, operating at 4K, multiple XPU systems will accelerate the highly parallel portions of the application(s). The XPUs consist of multiple RQL-based compute units (CU), both RQL and CMOS-based memories for cache, and memory controllers, which act as an interface to the storage needs of the system. The host processor, residing at warm space, is a conventional CMOS CPU. Both of the main-memories, the system and the device memory shown in Figure 1, are based on conventional CMOS technologies, such as DRAM HBM2, and reside in 233K. Router blocks, depicted as "R" in the figure, serve to route the data and control I/O between the different components.

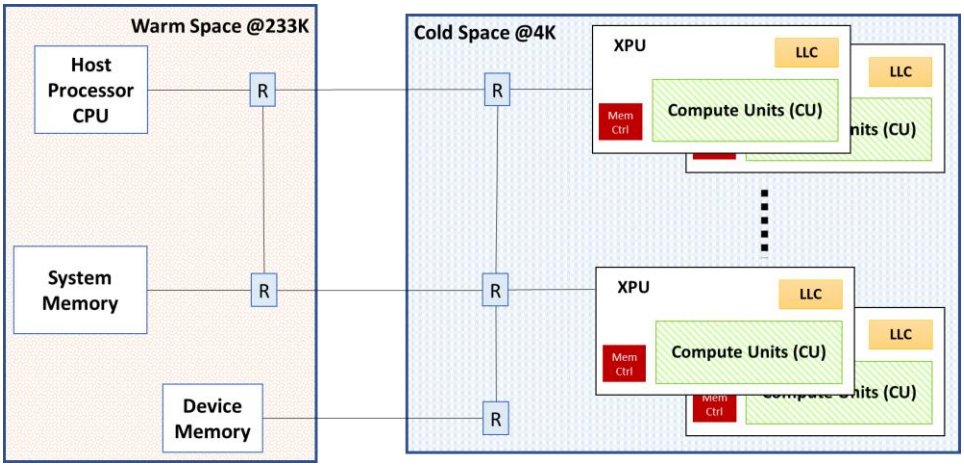

Figure 1: System Architecture.

For this work, the XPU is based off of the open-source MIAOW architecture which utilizes a subset of the Southern-Islands instruction set architecture from AMD (Balasubramanian et al. 2015). This architecture has a modular set of components that allowed us to optimize the accelerator to suit our specific application needs. Based on our target operating frequency of 10GHz, we calculate the per XPU throughput to be 5.12 TFLOPs for our system.

3 SIMULATION FRAMEWORK

A requirement for our system simulation environment is the ability to measure the impact of various components that affect the overall system architecture, including the impact of different memory and network configurations. To support our research goals, we selected the open-source simulation framework Structural Simulation Toolkit (SST) (Murphy 2007) from Sandia National Labs. This framework allows us to model the various components to suit the accuracy needs of our project. The framework is written in c++ with python configuration files used to describe the model of the system being simulated.

The specific components we used in our simulation environment are as follows: Ariel-Pin based CPU emulator, Shogun network simulator, simpleMem for the memory modeling, and GPGPUSim for modeling the accelerator, as illustrated in Figure 2. A dedicated command and data link coordinates the workload interface from the host processor to the XPU component(s). This coordination is accomplished using SST-Balar, an interface component between SST and GPGPUSim.

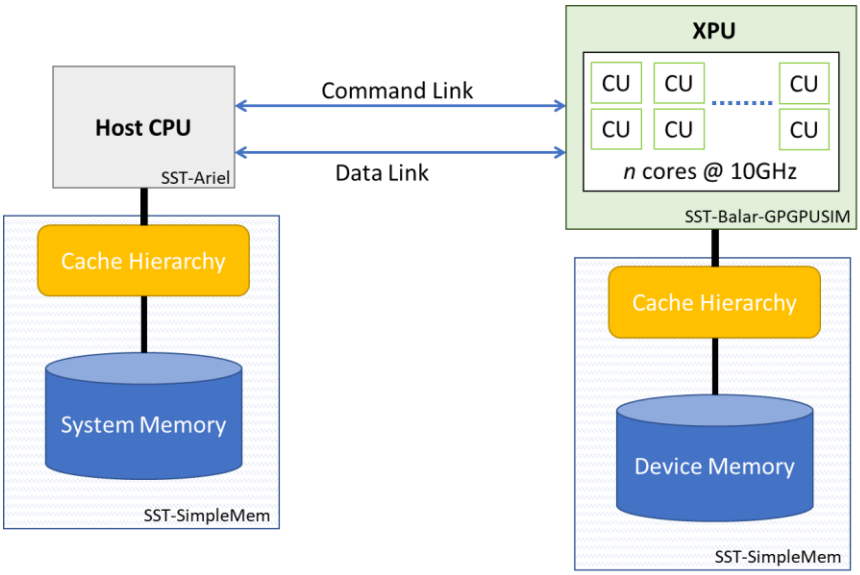

Figure 2: Simulation Framework.

Prior published results from Sandia modeled a NVIDIA V100 GPU, running at 1.3GHz, with an IBM Power-9 processor, and showed high correlation between hardware and model (Khairy et al. 2019). Since GPGPUSim models NVIDIA based systems, our work involved translating the XPU parameters, which are based on AMD terminology from the MIAOW architecture, into NVIDIA terminology used by the simulator. Some of the parameters of critical importance are the # of cores, clock-frequency, the initiation and pipeline latency for the different functional units, instruction buffer size, and register count. The latency parameters were matched to expected pipeline latency for the various instructions.

4 MODEL VALIDATION

As a first step, we validated the simulation environment and the model. Our method was to create a compute-intensive CUDA kernel to compare the pure computation performance as a way of baselining the system model. We compared our simulated results against hand-calculated values for the compute intensive kernel, consisting of a set of 32-bit add and other arithmetic operations executed in a loop. This kernel would remove any impact of the cache, memory load or store operations and also any effects from the network and would provide a first-order validation of the measured execution time.

Simulation statistics of the kernel showed 98% of the instructions were compute and less than 1% for both branch & memory operations. This particular "synthetic" kernel's sole purpose is to validate the model and is not intended to be a realistic representative benchmark. The loop count is varied to see the effect of the input dataset size on the throughput. Figure 3 shows a plot of the expected execution time from such a kernel, accounting for the loop overhead, as a function of the dataset size. As can be expected, the overall execution time increases linearly with large dataset size. There is an interesting effect at small dataset size, as the granularity of the modeled system dictates a step-size increase of the execution time that is observed. This can be seen in plot on the left in Figure 3.

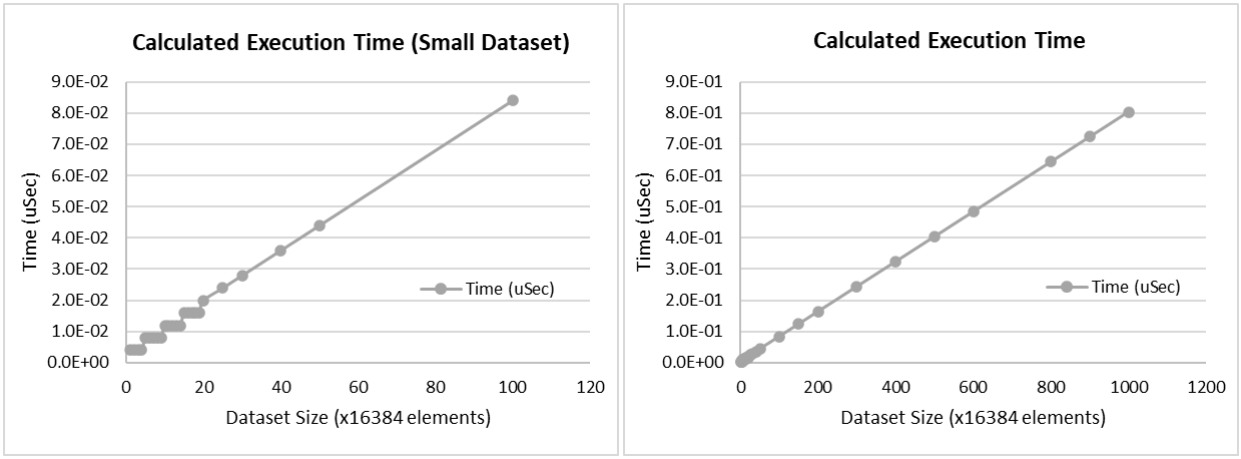

Figure 3: Calculated Execution Time at small (left) and large dataset size (right).

Using the calculated execution time across the range of dataset sizes, we calculated the expected throughput for 32-bit floating-point operations in our system. The results are shown in Figure 4. Hand-calculation shows that at small dataset sizes, we are well-below the peak throughput of the system, but as we increase the dataset size, we approach the max throughput of 20-TFLOPS. This is expected because the fixed overhead latency per instruction is amortized over the larger dataset size. Only at very large workloads, is the kernel able to fully exploit the throughput capabilities of the system.

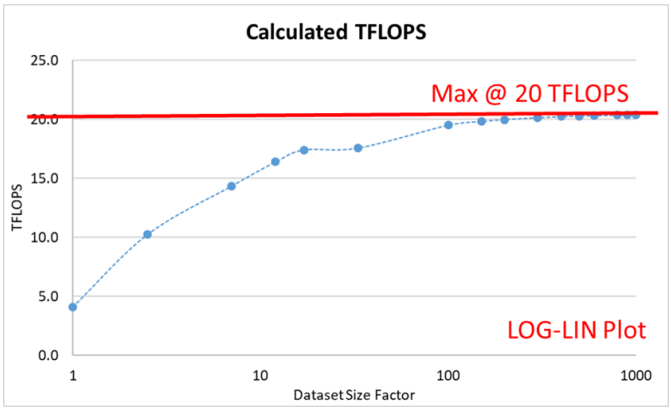

Figure 4: Calculated Throughput on Compute-Only Kernel.

We next simulated the compute-intensive kernel in the SST framework and compared with our expected throughput (shown in Figure 5). Results from SST-GPGPUSim simulations also point to a similar trend where, at smaller problem size, the throughput is well below the peak amount of the system, but as the problem size is increased, we approach the peak throughput. As shown in the figure, the simulated throughput matched well with the calculated throughput for the dataset size range that was simulated.

These results confirmed that our SST-GPGPUSim framework, from a computation throughput point of view, is modeling what we are expecting.

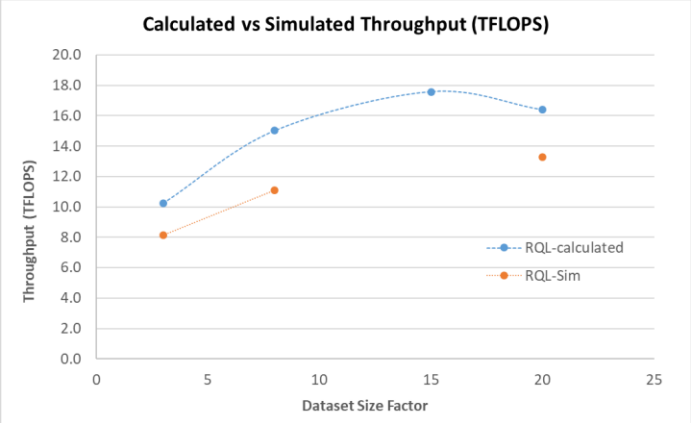

Figure 5: Comparison of Calculated against Measured Throughput on Compute-Only Kernel.

In addition to the hand-calculated correlation, we also compared the reported kernel performance from SST-GPGPUSim against a Gem5 simulation modeling a similar system for a FFT kernel from the SHOC benchmark (Danalis 2010). For the purpose of this simulation, we used the open-source AMD GCN3 model which utilized a 4-Compute Unit, 2MB Cache running at 1GHz (Butko et al. 2012). The results are shown in Figure 6.

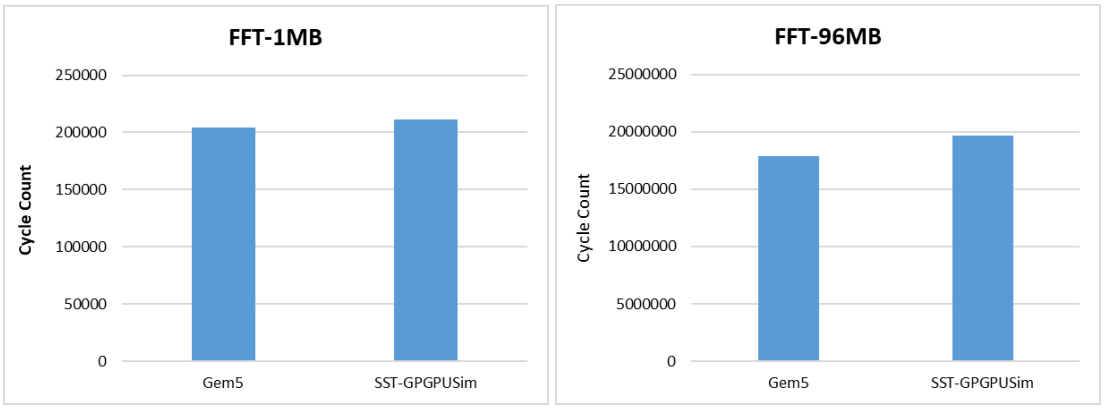

Figure 6: Correlation of SST-GPGPUSim result against Gem5 on an FFT kernel.

For our Gem5 correlation, we varied the FFT workload from a very small dataset size of 1MB to a very large dataset size of 96MB. The y-axis in Figure 6 shows the total kernel cycle count reported for both simulators. In both scenarios, the SST-GPGPUSim results tracked well with the Gem5 result for the FFT kernel.

Table 1 has the summary of the results, showing correlation of 4-10% of the results between the two simulators.

Table 1: SST-GPGPUSim to Gem5 Comparison.

Kernel Cycle Count	**FFT-1MB**	**FFT-96MB**
Gem5	203953	17871415
SST- GPGPUSim	211262	19667857
Correlation	3.58%	10.05%

5 SIMULATION STUDIES

Our first objective was to compare a baseline CMOS system against an RQL-based XPU accelerator system and evaluate the performance on a 1D Fast Fourier Transform (FFT) kernel. The FFT kernel is commonly used in most signal processing applications, and therefore was used as an initial system evaluation benchmark. The system model we used to make the comparison is shown in Figure 7.

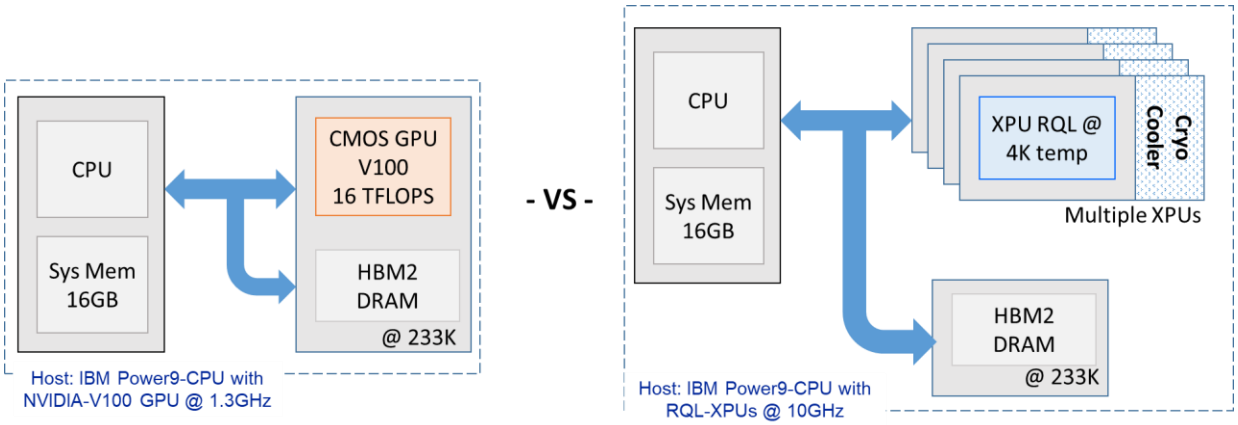

Figure 7: Baseline Comparison of CMOS vs RQL-XPU based System.

In order to assess the effect of the accelerator, we have kept all other components the same between the two models. Both systems use an IBM Power-9 CPU for the host processor with 16GB system memory and 16GB DRAM memory. On the CMOS-only model, the accelerator is a NVIDIA V100 system (Martineau et al. 2018), operating at 1.3GHz. The quoted FP-32 peak throughput for an NVIDIA V100 is 15.7 TFLOPs. For the RQL-XPU model, the accelerator is operating at a frequency of 10GHz. The figure also shows an attached CryoCooler component in the RQL system that will contribute to some of the power dissipation when we compare the power-efficiency of the two systems. While the exact cooling technology will determine the cooling cost, for this initial study we used a cooling cost of 300W/W for a typical liquid Helium cooled system at a temperature of 4K (Iwasa 2010).

Additionally, to study the impact of scaling, we increased the number of XPU accelerators. As we scaled the number of XPUs, we also scaled the last-level-cache capacity (LLC), the device memory, and the number of memory controllers (memCtrl) for the accelerator system. While having a high number of XPUs will improve the throughput of the system, the increase in power-efficiency will also be considered. The summary of the system scaling study configuration is given in Table 2.

Table 2: RQL-XPU System Parameters for Scaling Study.

	CMOS: V100	**XPU-1**	**XPU-2**	**XPU-4**	**XPU-8**	**XPU-16**
Compute-Unit Cores	84	8	16	32	64	128
LLC (MB)	6	10	20	40	80	160
Device Memory (MB)	16384	4096	8192	16384	32768	65536
memCtrl	8	2	4	8	16	32
Peak FP-32 Throughput (GFLOPS)	15670	6400	11520	21760	42240	83200

The first column describes the relative GPU parameters for the CMOS-V100 system used as the baseline. For the remaining columns, the RQL-XPU accelerator parameters are provided, where XPU-1 represents a

system containing a single XPU of 8 compute-units each. The LLC spec represents the sum-total of all of XPU's last-level-cache capacities for the system, where we used 10MB of LLC for each XPU accelerator.

5.1 Baseline Comparison and System Scaling Study

The RQL-XPU accelerator system and the memory model for the scaling study in the SST environment is shown in Figure 8. Each XPU is grouped with 8 compute-unit (CU) cores, and 10MB of last-level cache. The XPUs share the device memory through a memory bus, that scales with the number of XPUs. For this study, we used a device-memory scale factor of 4GB of device memory per XPU, as shown in the previous table.

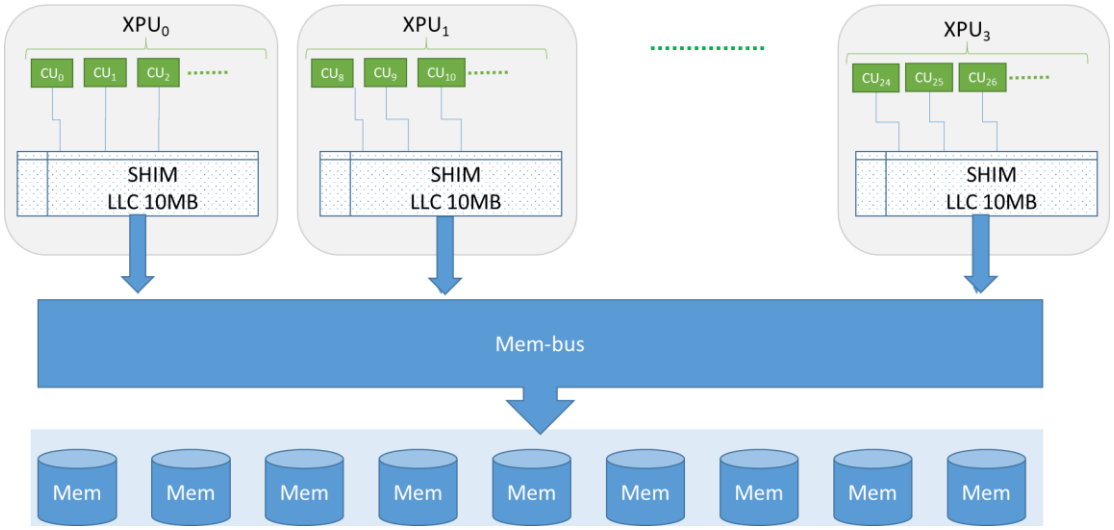

Figure 8: System and Memory Architecture for RQL Scaling Study.

The results for a small FFT kernel with 1MB dataset size are shown in Figure 9. The plot on the left compares the execution time for a CMOS-V100 system against RQL-XPU system of various sizes. For this kernel, we observe that a single XPU system will be outperformed but increasing to even just a two-XPU system improves the kernel execution time by a factor of 2. We also observe a leveling in the performance as we continue to scale up with the plot on the right. This plot shows the speedup of the kernel execution time over a single-XPU system, along with the speedup-efficiency (secondary-axis on the right). Speedup-Efficiency is defined as the speedup divided by the number of XPUs (max of 1). The results show that for this kernel and dataset size, the optimum XPU count is between 4 (32 CUs) and 8 (64 CUs). Beyond this point, the overall increase in power consumption may not provide additional benefits.

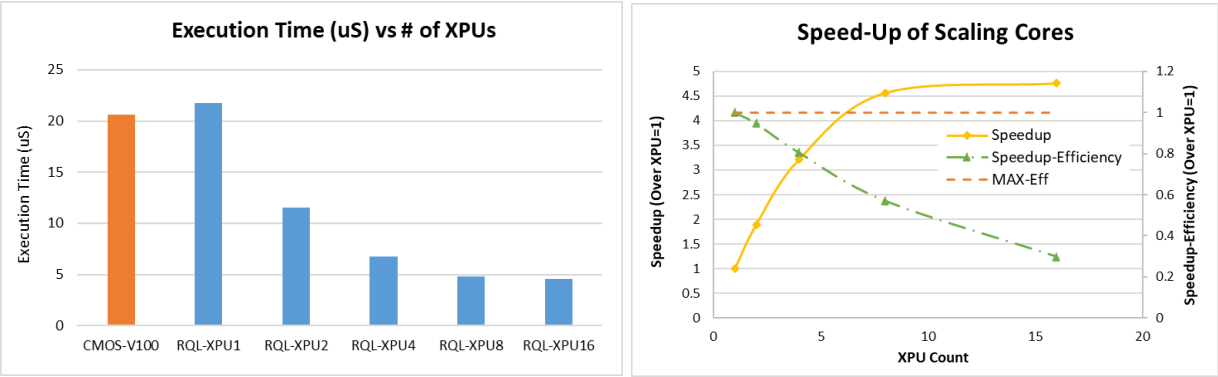

Figure 9: System Scaling Study for FFT-1MB testcase.

The summary of the expected power-consumption numbers is presented in Table 3 for both the baseline CMOS-V100 system and an RQL system with varying number of XPUs. Power numbers are calculated from published data for the CMOS components and manually computed based on the total number of Joseph Junctions for the RQL components. The power numbers include the cooling costs to maintain the system at either 4K or 233K, at a cost of 300W/W or 1.2W/W, respectively. Since the overhead chips, the host-processor and main-memory, are located in the warmer temperature regions, they contribute to much more of system power than the per-blade RQL power. As can be seen in the table, the large CPU and DRAM memory power consumption is better amortized by the accelerator or GPU power at larger scales. However, this comes with a cost of increased overall system power and implementation considerations, which we will discuss in the next section.

Table 3: System Power, Throughput, and Power-Efficiency.

	CMOS-V100	**RQL-XPU1**	**RQL-XPU2**	**RQL-XPU4**	**RQL-XPU8**	**RQL-XPU16**
Throughput (GFLOPS)	15670	6400	11520	21760	42240	83200
CPU+DRAM Power on 233K side(W)	239.4	233.7	235.6	239.4	247	262.1
Accelerator Power on 4K (W)	250	17.1	34.2	68.4	136.8	273.6
System Power (W)	489.4	250.8	269.8	307.8	383.8	535.7
Power-Eff (GFLOPS/W)	32.02	25.52	42.70	70.70	110.06	155.31

Next, we expanded the scaling study to a larger FFT dataset size of 96MB. The simulation results are shown in Figure 10. The plot shows that as we increase the number of XPUs, the execution time continues to lower, and does not level-off as it did with the smaller dataset size. This indicates that this larger dataset size more fully utilizes the additional compute resources. We also observe that with the larger dataset size, the single CMOS-V100 system is able to outperform RQL-XPU for both the single and two-XPU systems. This could be explained by the peak throughput of the XPU1 and XPU2 in comparison with the CMOS-V100 system, as shown in Table 3.

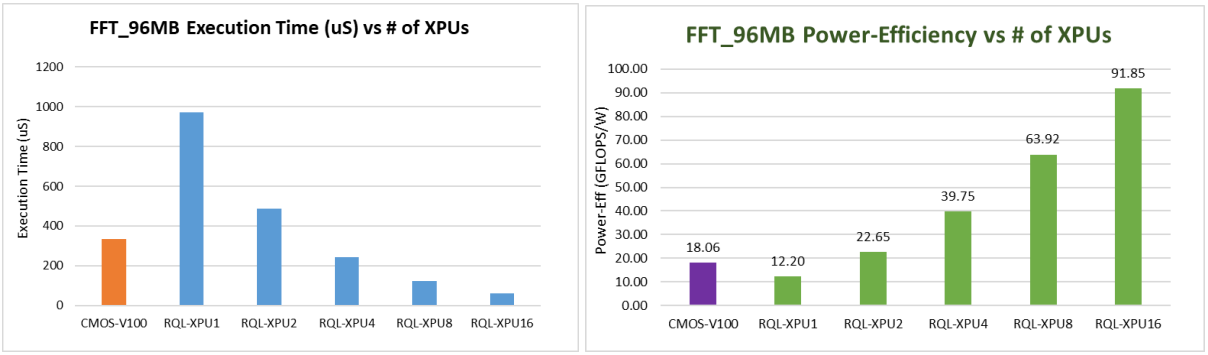

Figure 10: System Scaling Study of FFT-96 MB case.

The second plot in Figure 10 shows the Floating-Point-32 bit specific power-efficiency of the system using the following throughput formula for a 1D FFT kernel, where N represents the number of elements simulated and t represents the execution time.

$$FFT\ Throughput = \frac{5N \log_2 N}{t}$$

From the power-efficiency plot, we see that other than the single XPU system, all XPU count outperform the CMOS-V100 system in terms of power-efficiency, with the overall power efficiency increasing as the XPU count is scaled up. Two factors create an upper limit on the number of XPUs – the first is the raw system power, as summarized in Table 3. As we increase the number of XPUs, the contribution from the power of the accelerator might exceed the system power budget. The second factor is a physical limitation on the number of XPUs that can be connected to a single host-processor. This comes from both the physical connection limit, as well as the capability of what the host-processor can handle to support the parsing of the workload and handling serial tasks from the larger application.

5.2 Memory Parameter Study

We also conducted an experiment to study the impact of the main-memory controller and the request buffer width on the overall system performance for a 4-XPU system. While having more memory controllers will help with removing contention, as seen in the results in Figure 11, there does seem to be a limit to the return on performance with adding beyond 4 memory controllers for a 32-CU or 4-XPU system. Therefore, any performance benefit would need to be balanced with the additional power contribution, especially at the 4K temperature region.

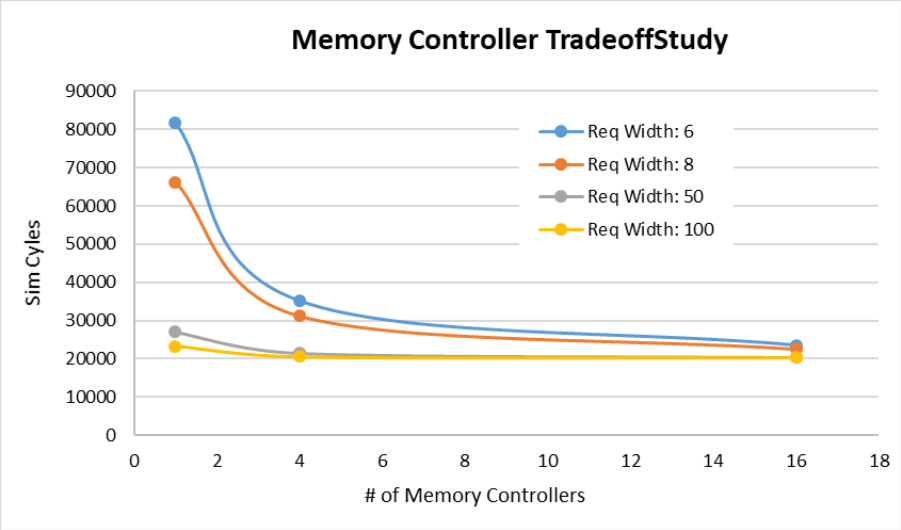

Figure 11: Memory Controller and Request Width Study.

We also varied the memory controller's input request width from 6, 8, 50, and 100. For our system, a memory input request width of 50 seemed sufficient in terms of improving system performance for low memory controller scenario. However, as we increased the number of memory controllers, the request width's impact on system performance decreased. This is likely due to the increase in memory controllers providing additional memory bandwidth and alleviating contentions, thereby reducing the need for a large request buffer. Further study is needed to optimize on both of these memory parameters on the system performance.

5.3 System Comparison on Representative Kernels

For our final study, we compared the system model on four different kernels– FFT, Sparse Matrix Vector Multiplication (SPMV), Lower-Upper Decomposition (LUD), and Discrete Wavelet Transform (DWT). These kernels, commonly used in signal processing applications, were chosen to give an indication of

performance on spectral processing and sparse and dense linear algebra methods. We first evaluated the impact of the pipeline latency on the overall system performance, in comparison with the CMOS-V100 system. For this study, we focused on the RQL-XPU system with 4 XPU configurations, which performed optimally based on the previous simulation studies. We varied the pipeline latency from *n* cycles per FP-32 instruction to *2.2n* cycles per FP-32 instruction, based on physical implementation constraints. The result for this study is presented in Figure 12, normalized to allow for comparison across the various kernels with different workloads. As seen in the figure, the RQL system performed worse than CMOS on the XPU-4 (32-CU) system for the FFT and DWT kernels. However, a lowered pipeline latency of *n* cycles per FP-32 instructions allows for the performance advantage across all four kernels.

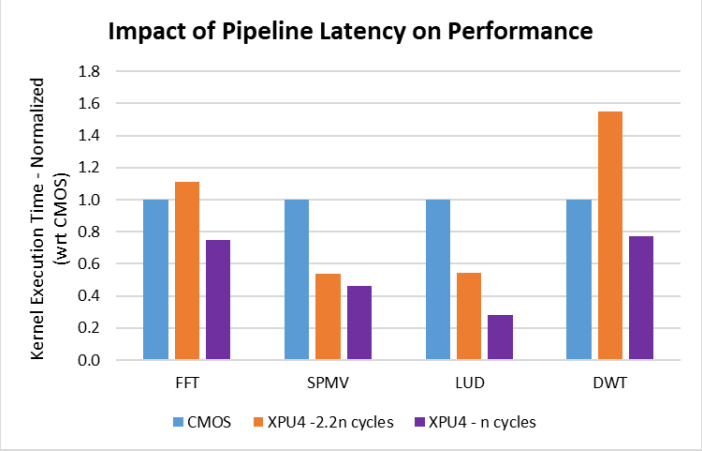

Figure 12: Impact of Pipeline Latency on System Performance.

We also evaluated an XPU-8 (64-CU) system with the higher pipeline latency of *2.2n* cycles per FP-32 instruction. The results, shown in Figure 13, indicate that increasing to 8 XPUs increases performance. However, as mentioned earlier, this comes at a cost of increased power that might need to be mitigated. To illustrate the impact of the additional power contribution against the increase system performance that an XPU-8 system provides, the second plot in Figure 13 provides a power-delay-product.

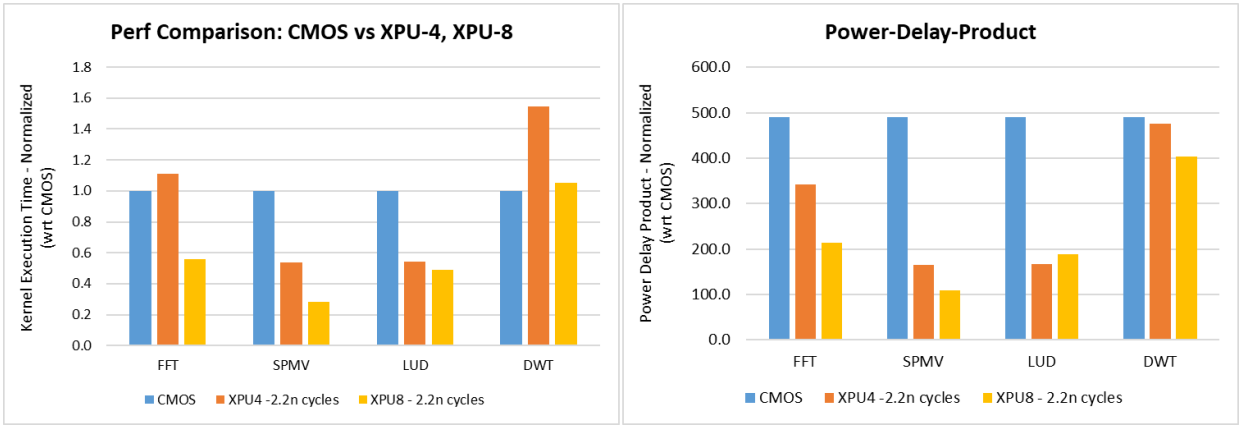

Figure 13: Performance and Power-Delay-Product Comparison: CMOS vs XPU-4, XPU-8.

The power-delay product is computed by multiplying the overall system power (W) with the execution time (uS), as a rough first-order comparison. Using this criterion, scaling up to XPU-8 does provide sufficient increase in performance to overcome the power cost for 3 out of the 4 kernels we evaluated – FFT, SPMV, and DWT. LUD had a marginal increase of power-delay-product, which indicates that the improvement in performance is not as significant as the increase in power. While we used a weight of 1 for both power and

delay, this formula can be weighed different to prioritize these two parameters differently based on the application priorities.

5.4 Performance At-Scale and Future Direction

One of the key advantages of the superconducting RQL technology is the scalability to deliver high amount of compute in a small volumetric-form factor. This is due to the fact that due to unique properties of superconducting electronics, at cryogenic (4K) temperatures, scaling the system for increased compute capability increases the overall system volume at a much slower rate than for conventional CMOS circuits. For example, a 1600-XPU system is extrapolated to be capable of delivering over 8 peta-flops of throughput with a very small form factor and maintain two orders of magnitude higher volumetric power efficiency. Future work intends to study the behavior of the system at these larger scales, including the impact of network configurations.

On the memory architecture side, from a physical implementation point of view it would be advantageous to combine the host and device memory together, since both (in DRAM) are planned to reside in the 233K region. This unified memory architecture might make sense especially for sub-groups of the system designed to be allocated to a single CPU (ex: 1 CPU to 8 XPUs sharing main-memory). Some industry SOCs are already moving towards this approach to save on memory copy times. NVIDIA's unified architecture still requires a copy to a different location in the physical memory. We will continue our evaluation to assess if such an approach of having unified main-memory is feasible for our system.

6 CONCLUSION

In this paper, we presented a simulation framework and methodology to evaluate the system performance and power efficiency of a superconducting RQL based system consisting of accelerators based off the MIAOW architecture. We used a conventional CMOS system using NVIDIA-V100 for our baseline comparison against our superconducting system. Simulation results on different kernels of interest showed an overall performance and power-delay improvement of 41% and 53%, respectively for an 8-XPU RQL system with a peak throughput of 42 TFLOPs, with expected power efficiency benefits at large scale. Further studies on the impact of system parameters is needed to optimize the overall system parameter to suit the application needs.

REFERENCES

Balasubramanian, R., Gangadhar, V., Guo, Z., Ho, C.-H., Joseph, C., Menon, J., Drumond, M. P., Paul, R., Prasad, S., Valathol, P., & Sankaralingam, K. 2015. "Enabling gpgpu low-level hardware explorations with Miaow". In *ACM Transactions on Architecture and Code Optimization*, vol. 12(2).

Butko, A., Garibotti, R., Ost, L., & Sassatelli, G. 2012. "Accuracy evaluation of GEM5 simulator system". In *7th International Workshop on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC)*.

Byun, I., Min, D., Lee, G., Na, S., & Kim, J. 2020. "CryoCore: A Fast and Dense Processor Architecture for Cryogenic Computing". In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*.

Danalis, A., Marin, G., McCurdy, C., Meredith, J. S., Roth, P. C., Spafford, K., Tipparaju, V., & Vetter, J. S. 2010. "The Scalable Heterogeneous Computing (SHOC) benchmark suite". In *Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units - GPGPU '10*.

ITRS, *The International Roadmap for Devices and Systems: 2021*.

Iwasa, Y. 2010. “Case studies in superconducting magnets: Design and operational issues”. In *New York: Springer*.

Khairy, M., Zhang, M., Green, R., Hammond, S.D., Hoekstra, R., Rogers, T., and Hughes, C. 2019. “SST-GPU: An Execution-Driven CUDA Kernel Scheduler and Streaming-Multiprocessor Compute Model”. In *Internal SAND Report SAND2019-1967, 2019.*

Lee, A., Davis, A., Ratz, P., Przybysz, A., Marakov, A., Medford, J., Pesetski, A., & Przybysz, J. 2021. “Finding and eliminating noise and interference in a test stand for Josephson digital chips”. In *IEEE Transactions on Applied Superconductivity*, vol. 31(5), pp. 1–5.

Manheimer, M. A. 2015. “Cryogenic computing complexity program: Phase 1 introduction”. In *IEEE Transactions on Applied Superconductivity*, vol. 25(3), pp. 1–4.

Martineau, M., Atkinson, P., & McIntosh-Smith, S. 2018. “Benchmarking the nvidia V100 GPU and tensor cores”. In *Lecture Notes in Computer Science*, pp. 444–455.

Murphy, R., Underwood, K., Rodrigues, A., and Kogge, P. 2007. “The Structural Simulation Toolkit: A Tool for Exploring Parallel Architectures and Applications”. In *Internal SAND Report SAND2007-0044C, 2007*.

AUTHOR BIOGRAPHIES

MEENATCHI JAGASIVAMANI is a Staff Systems Engineer at Northrop Grumman Corporation. She holds a PhD in Electrical and Computer Engineering from University of Maryland, College Park. Her research interests lie in architecting novel solutions using emerging technologies. Her email address is meenatchi.jagasivamani@ngc.com.

CHRISTINE FONG is a Sr. Principal Systems Architect at Northrop Grumman Corporation. She holds a Masters degree in Electrical and Computer Engineering from Carnegie Mellon University. Her research interests lie in system architecture and high-performance computing. Her email address is christine.fong@ngc.com.

KEN GOODNOW is a Sr. Consulting Engineer at Northrop Grumman Corporation. He holds a PhD in Electrical Engineering from Penn State University. His research interests lie in transformation computing, microelectronics and digital system design. His email address is kenneth.goodnow@ngc.com.

ROBERT VOIGT is a NG Fellow and Chief Technologist at Northrop Grumman Corporation. He holds a PhD in Electrical Engineering from Naval Postgraduate School. His research interests lie in transformational computing using superconducting electronics, artificial intelligence, and machine learning. His email address is robert.voigt@ngc.com.