A CUDA-QUDA architecture, Hyper-Data Quantum GPUs

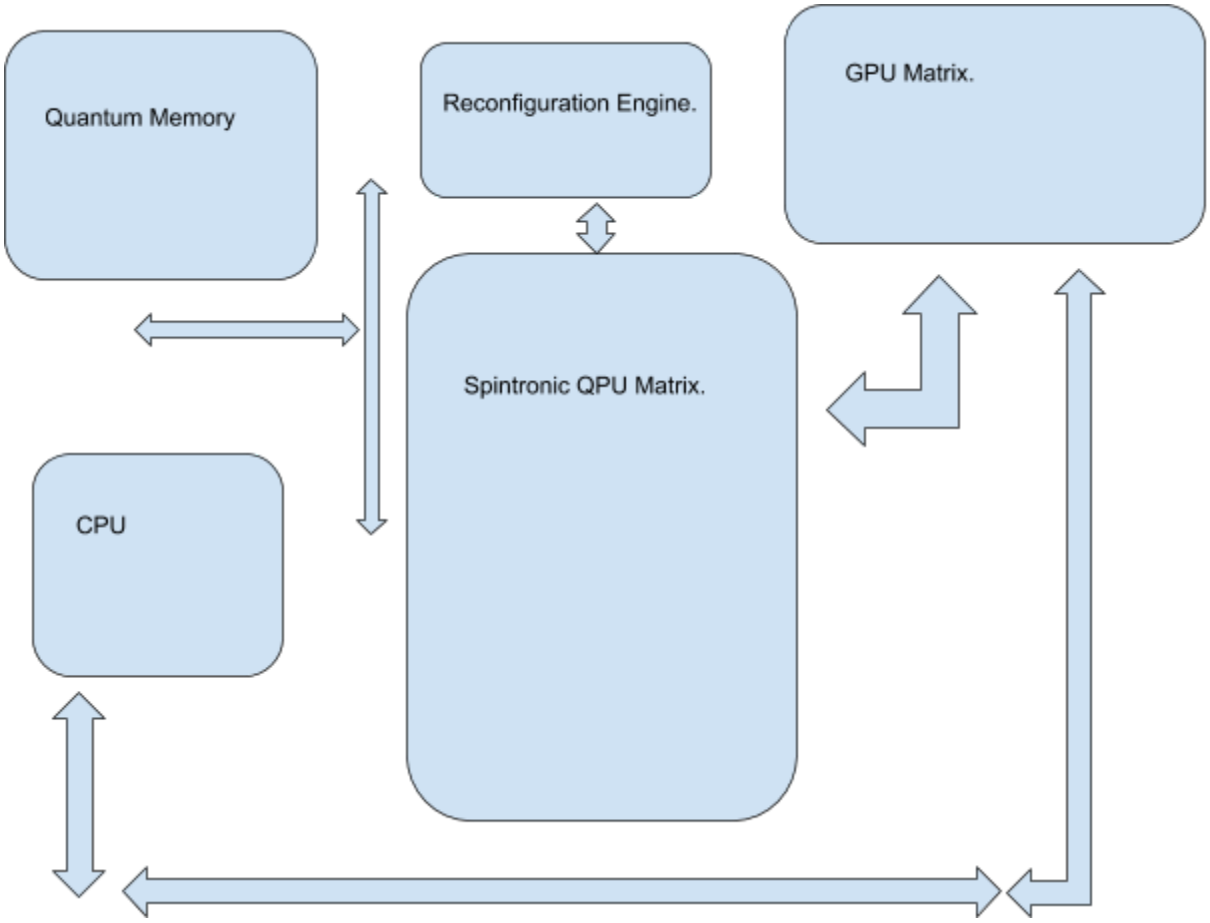
Dr Bheemaiah, Anil Kumar, A.B Seattle W.A 98125
bheemaiaha@yopmail.com

Abstract:
QUDA is an architecture similar to CUDA for HPC applications of Quantum GPU architectures to be used in conjunction with GPU and MCU based processing. There is no QUDA pipeline similar to a stream processing architecture or an out of order instruction pipeline. True ILP is achieved with a QUDA architecture, which is better than quasi-parallelism in a CUDA or scalar/vector architecture.

Keywords: QUDA architecture, HPC, Quantum Cloud, Qiskit, spin waves, spintronic, hyper-data, quantum operating system.

What:

Quantum Unified Compute Architecture or QUDA architectures is the use of Q and Qiskit for a reconfigurable quantum array architecture on spintronic units with a reconfigurable scheduler for a quantum operating system.



How:

We consider a QUDA architecture in python to simulate a reconfigurable QPU cell-matrix on the Q cloud in association with a CUDA kernel. We consider the existence of shared memory and ultra-fast hyper-data support, with all other CUDA functionalities and libraries supported.

Why:

A large body of efficient quantum algorithms for stream processing, vector data processing and hyper-data algorithms in Hilbert space exist for very efficient HPC computing. With time it is possible that it be proven that P and NP are actual subsets of QP with polynomial-time convergence and bounds proven for most HPC workloads, enabling serious QPU designs to supplement the CUDA architecture.
Low-temperature silicon or other non-linear materials like graphene at atomic scale fabrication will make inexpensive quantum integration possible leading to a revolution in 100 qubits + chip architectures as QPU units, to act as a coprocessor to MCU-GPU units. Thus QPU-MCU hybrid units unify with MCU-GPU-QPU hybrid units leading to the QUDA architecture.

Summary:

Main Points:

Adiabatic Ising spin-glass phase transition as a coherence model in multi-cell QPU architectures.

Spin Wave Transfer as data and control architecture.
Shared Memory and data prediction algorithms.
Reconfigurability as algorithms in QPU allocation.

Applications:

CUDA architectures allow for HPC workloads on GPU architectures, while several algorithms exist and are faster in QP, the classes QT on taskoid computability and several other algorithms including data mining workloads, allow for an expert system or heuristic-based work scheduling between GPU/QPU controlled by MCU resident routines.
Thus QPU-GPU-MCU architecture is HPC, leading to the QUDA architecture.

Code Base:

Github:

Introduction.

Quantum GPU architectures have been defined in an earlier publication (Bheemaiah).In this paper, we define a CUDA inspired quantum architecture called QUDA, with a dynamic architecture for a quantum cloud architecture similar to IBM Q Experience.

In this paper, we define some formal definitions. In a future paper, we define the theorems for a formal analysis of the design of such a system and a practical simulation in python similar to Qiskit.

Problem Definition.

QUDA architectures are defined as analogous to CUDA architectures(Ghorpade 2012; Nickolls 2007; Gulati and Khatri 2010; Farber 2011) with GPU-QPU-MCU architectures, with global memory and a SIMD inspired architecture,

1. Micro Hyperspace -Data: DM and ZDM and QDM define the spaces for computability of memory streams mined for patterns, metadata, and sequences used in caching and prediction in reconfigurability.
2. Reconfigurability is defined over the use of QPU units as the QUDA architecture for GP HPC.
3. [R] is the scheduling of reconfigurability in threads and execution on the QPU matrix, by an expert system on QP algorithms, a subject of a future publication.

Background.

<original-contribution>

Formal Definitions:

Hyper-Data(Wittek 2014) is an application of data-mining, sequence prediction, and pattern metadata, in dynamic architectures as reconfigurability.
Many architectures in classical CUDA and superscalar architectures exist.
Definition 1.0 : Consider a QPU-GPU-MCU architecture.
In a CUDA-QUDA formulation, a matrix of GPU, [[G_i_j]] and QPU [[Q_i_j]] with memory, [[M_i_j]] and an MCU M, caches, [C_i], reconfigurabilty [R].
Definition 1.1 : Hyper-data is defined as micro or macro hyper-scale data. In microscale hyper-data, data-mined architectures,
Tr -> ([[M_i_j]], R), where QDM, is the computability of QP algorithms for data mining memory streams, for pattern clusters, metadata and sequence prediction. There can also exist DM and ZDM in P and NP for classical and stochastic algorithms, executable in [[G_i_j]].
Definition 1.2 : STT is the quantization of coherence in MCU-QPU and MCU-GPU-QPU coherences with shared memory, global memory coherence in Hilbert space over [[M_i_j]], in $\Sigma\Sigma\psi ij()$ (recap: Q.M is linear and state-based with wave functions in a probabilistic framework with observables and operators)
There exist phase transitions, p_i_t(), where a transition [s_i] exists of a finite number of coherences in spin-torque transfer, to quantum states which can act as input and output, analogous to read-ins and readouts to each individual QPU units spin buffer sb. This coherence forms the analog of stream inputs and outputs on QPUs replacing classical bus-based architectures.

</original-contribution>

Discussion.

True ILP is achieved with a QUDA architecture, which is better than quasi-parallelism in a CUDA or scalar/vector architecture.
Adiabatic leads to a better than vector/scalar distribution of reconfigurability in ILP for MCU-GPU thread scheduling, asynchronously. This leads to better models than the CUDA architecture and GPU

pipelines and out or order pipelines in brute force multi-core implementations of scalar architectures.
We have presented dynamic scheduling of QPU processing from causal processing of threads, based on heuristics for the use of QPU units for general processing, called the QUDA architecture.

Future Work.

[R] is the scheduling of reconfigurability in threads and execution on the QPU matrix, by an expert system on QP algorithms, a subject of a future publication.
A simulator similar to Qiskit in python (“Qiskit” n.d.)is the topic of future research.

References.

Farber, Rob. 2011. *CUDA Application Design and Development*. Elsevier.

Ghorpade, Jayshree. 2012. “GPGPU Processing in CUDA Architecture.” *Advanced Computing: An International Journal*. https://doi.org/10.5121/acij.2012.3109.

Gulati, Kanupriya, and Sunil P. Khatri. 2010. “GPU Architecture and the CUDA Programming Model.” *Hardware Acceleration of EDA Algorithms*. https://doi.org/10.1007/978-1-4419-0944-2_3.

Nickolls, John. 2007. “GPU Parallel Computing Architecture and CUDA Programming Model.” *2007 IEEE Hot Chips 19 Symposium (HCS)*. https://doi.org/10.1109/hotchips.2007.7482491.

“Qiskit.” n.d. Accessed August 26, 2019. https://qiskit.org.

Wittek, Peter. 2014. *Quantum Machine Learning: What Quantum Computing Means to Data Mining*.