Towards Complete Emulation of Quantum Algorithms using High-Performance Reconfigurable Computing

Naveed Mahmud

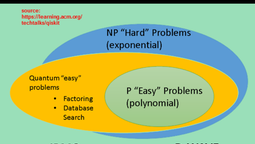
Department of Electrical Engineering and Computer Science, University of Kansas

naveed_923@ku.edu

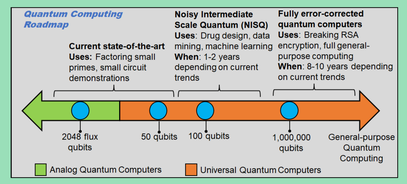
Ph.D. Advisor: Dr. Esam El-Araby

Introduction and Motivations

- Why Quantum?
  - Solving NP-hard problems
  - Speedup over classical



- Critical Problems
  - I/O intensive applications
  - Deep, complex circuits
  - High cost of access
  - Noisy and Intermediate Scale
  - Verification and Benchmarking



- Emulation using FPGAs
  - Greater speedup vs software
  - Dynamic (reconfigurable) vs. fixed architectures
  - Exploiting parallelism
  - Limitation → Scalability

- Problems with existing Simulators/Emulators
  - Costly, resource-hungry, and power-hungry SW simulation platforms
  - Poor scalability, low precision, and low throughput HW emulators
- Contributions
  - A cost-effective, High-Level Synthesis (HLS) and FPGA accelerated methodology for complete emulation of quantum algorithms
  - Depth optimizations for classical-to-quantum (C2Q) data encoding and Quantum Haar Transform (QHT)
  - Extension of Quantum Grover's search (QGS) for dynamic, multi-pattern search
  - Combining QHT and multi-pattern QGS to perform dimension reduction for dynamic pattern recognition on high-resolution, spatio-spectral data

Background

- Quantum bit (qubit)
  - Smallest unit of computation
  - Can exist in superimposed state:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

- Quantum gates
  - Unitary transformations on qubits
- Quantum circuit depth
  - Number of gates in longest circuit path

Hadamard Gate — $H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$

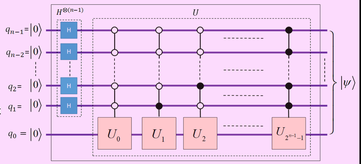
Controlled NOT — $CNOT = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$

Y-Rotation Gate — $R_y(\theta) = \begin{bmatrix}\cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right)\\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right)\end{bmatrix}$

Z-Rotation Gate — $R_z(\varphi) = \begin{bmatrix}e^{-\frac{i\varphi}{2}} & 0\\ 0 & e^{\frac{i\varphi}{2}}\end{bmatrix}$

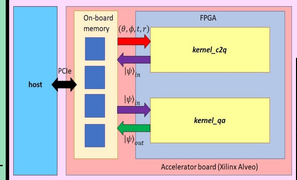
Classical-to-Quantum

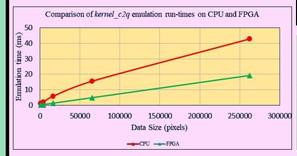
- Classical-to-quantum (C2Q)
  - Synthesize arbitrary quantum state from ground state qubits
  - Uniformly-controlled operations **(U_j)** + Hadamard **(H)** gates
  - Parameters for U_j extracted using data coefficient pairs from input data set



- Emulation Platform
  - 16-core, 3 GHz AMD EPYC CPU with 251 GB system memory
  - Xilinx Alveo accelerator board with xcu250 FPGA
  - Vitis unified software platform

$$U_j = R_z(\phi_j)\cdot R_y(\theta_j)\cdot r_j e^{i\left(\frac{\varphi_j}{2}\right)}$$

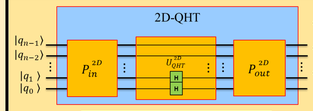




	Number of data items (N)	Number of qubits (n)	Simulation (MATLAB / ibm_qasm) Gate Count	Simulation (MATLAB / ibm_qasm) Circuit Depth	Implementation (ibmq_16_melbourne) Gate Count	Implementation (ibmq_16_melbourne) Circuit Depth
Complex randomized data	4.00E+00	2	7.00E+00	6.00E+00	1.70E+01	1.50E+01
	1.60E+01	4	3.30E+01	3.00E+01	9.20E+01	7.20E+01
	6.40E+01	6	1.31E+02	1.26E+02	3.38E+02	2.76E+02
	2.56E+02	8	5.17E+02	5.10E+02	NA due to hardware limitations of IBM-Q	
	1.02E+03	10	2.06E+03	2.05E+03		
	4.10E+03	12	8.20E+03	8.19E+03		
	1.64E+04	14	3.28E+04	3.28E+04		
Image data	2 x 2 pixels	2	3.00E+00	2.00E+00	1.30E+01	1.10E+01
	4 x 4 pixels	4	1.70E+01	1.40E+01	5.80E+01	4.70E+01
	8 x 8 pixels	6	6.70E+01	6.20E+01	2.38E+02	1.97E+02
	16 x 16 pixels	8	2.61E+02	2.54E+02	8.71E+02	7.46E+02
	32 x 32 pixels	10	1.03E+03	1.02E+03	NA due to hardware limitations of IBM-Q	
	64 x 64 pixels	12	4.11E+03	4.09E+03		
	128 x 128 pixels	14	1.64E+04	1.64E+04		

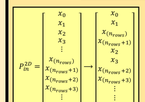
Data size (no. of pixels)	No. of states (N)	No. of qubits (n)	*kernel_c2q* emulation time (ms) CPU	*kernel_c2q* emulation time (ms) FPGA	*kernel_qht* emulation time (ms) CPU	*kernel_qht* emulation time (ms) FPGA	Total emulation time (ms) CPU	Total emulation time (ms) FPGA	Speedup
16x16x3	1024	10	0.98	0.081	0.464	0.006	1.444	0.087	16.60
32x32x3	4096	12	1.879	0.304	0.526	0.014	2.405	0.318	7.56
64x64x3	16384	14	5.696	1.204	0.527	0.031	6.223	1.235	5.04
128x128x3	65536	16	15.483	4.813	0.669	0.118	16.15	4.931	3.28
256x256x3	262144	18	42.776	19.232	1.003	0.441	43.78	19.67	2.23

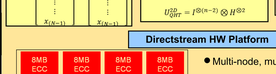
Quantum Haar Transform

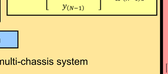


- Input permutations
- Haar kernel
- Output permutations

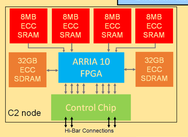
2D quantum Haar transform circuit





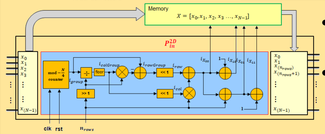


Directstream HW Platform

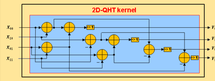


- Multi-node, multi-chassis system
- FPGA-only (Intel Arria 10) compute node
- 4x8MB SRAM and 2x32GB SDRAM on-board memory
- HLL-based hardware development

Emulation HW architectures



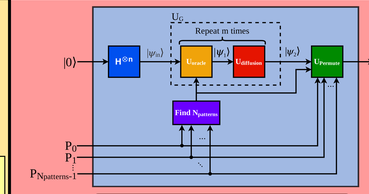
- Input permutations modeled as a hardware scheduler.
- Output vector elements read from memory



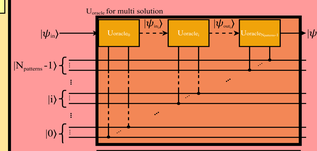
- Hardware kernel for 2D Haar operation
- Models the Haar operation with simplified shift and addition operations

Number of qubits, n	Number of pixels	On-chip[a] resource utilization (%) ALMs	BRAMs	DSPs	OBM utilization (bytes) SDRAM	Emulation time (sec)
8	16x16	14	9	2	4K	6.73E-06
10	32x32	14	9	2	16K	1.66E-05
12	64x64	14	9	2	64K	5.62E-05
14	128x128	14	9	2	256K	0.0002
16	256x256	14	9	2	1M	0.0008
18	512x512	14	9	2	4M	0.0033
20	1024x1024	14	9	2	16M	0.0135
22	2048x2048	14	9	2	64M	0.0540
24	4096x4096	14	9	2	256M	0.2160
26	8192x8192	14	9	2	1G	0.8641
28	16Kx16K	14	9	2	4G	3.4563
30	32Kx32K	14	9	2	16G	13.825

Modified Grover's Search

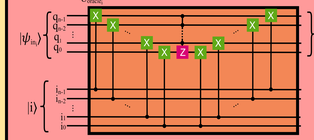


Proposed/modified multi-pattern Grover's search



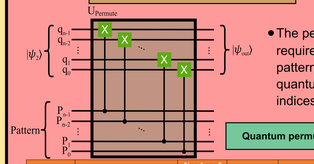
- The multi-pattern oracle is constructed of cascaded single-pattern oracle circuits

Modified oracle for Grover's algorithm for multiple solutions/patterns



- Each oracle also utilizes controlled Pauli-X (cX) gates instead of Pauli-X gates to allow dynamic search patterns

Modified oracle for Grover's algorithm for a single solution/pattern



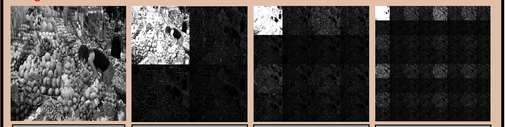
- The permutation step is required to shift the target patterns to the output quantum register's target indices

Quantum permutation circuit

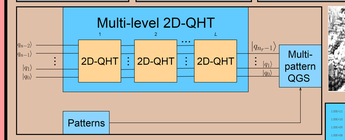
Simulation Platform	Reported work	Number of search patterns	Number of qubits	Precision type	Frequency (MHz)
CPU	Avila et al. [11], 2017	single	21	32-bit floating pt.	3400
GPU	Avila et al. [11], 2017	single	21	-	1000
	Gutiérrez et al. [12], 2010	single	26	32-bit floating pt.	1350
FPGA	Khalid et al. [13], 2004	single	3	16-bit fixed pt.	82.1
	Lee et al. [14], 2016	single	7	24-bit fixed pt.	85
	Proposed Work	single/	32	32-bit floating pt.	233

Dimension Reduction and Pattern Recognition and Matching

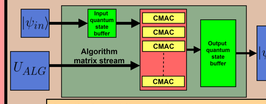
- High-resolution test images up to 64k × 64k pixels
- Multi-level packet 2D-QHT → L-level decomposition to target resolution
- 32-qubit 2D-QHT and 16-qubit Grover's search circuits implemented on a single FPGA



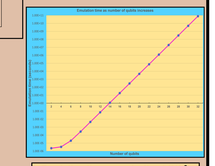
Original 512 × 512 image | 256 × 256 images after 1-level 2D-QHT | 128 × 128 images after 2-level 2D-QHT | 64 × 64 images after 3-level 2D-QHT



Pattern recognition and matching using Multi-pattern QGS



Architecture of the quantum algorithm emulator



Emulation time complexity of $O(N^2)$

Comparative Study

Reported Work	Algorithm	Number of qubits	Precision	Operating frequency (MHz)	Emulation time (sec)
Fujishima (2003)	Shor's factoring	-	-	80	10
Khalid et al (2004)	QFT	3	16-bit fixed pt.	82.1	61E-9
	Grover's search	3	16-bit fixed pt.		84E-9
Aminian et al (2008)	QFT	3	16-bit fixed pt.	131.3	46E-9
Lee et al (2016)	QFT	5	24-bit fixed pt.	90	219E-9
	Grover's search	7	24-bit fixed pt.	85	96.8E-9
Silva and Zabaleta (2017)	QFT	4	32-bit floating pt.	-	4E-6
Pilch and Długopolski (2018)	Deutsch	2	-	-	-
Proposed work	**QFT**	**32**	**32-bit floating pt.**	**233**	**7.92E10**
	QHT	**30**			**13.825**
	Grover's search	**32**			**7.92E10**
	QHT + Grover's	**32**			**7.92E10**

Conclusion

- Efficient and cost-effective emulation/simulations methods required for quantum computing technology
- An FPGA-accelerated methodology is proposed for complete emulation of quantum algorithms
- Higher scalability, throughput and accuracy vs existing HW emulators