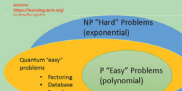
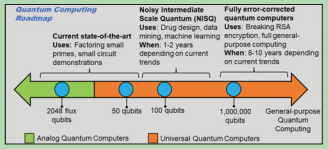
Towards Complete Emulation of Quantum Algorithms using High-Performance Reconfigurable Computing

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



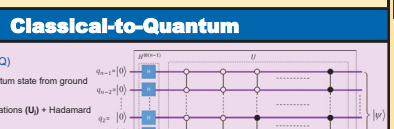


- **Emulation using FPGAs**
  - Greater speedup vs software
  - Dynamic (reconfigurable) vs. fixed architectures
  - Exploiting parallelism
  - Limitation → Scalability

- **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

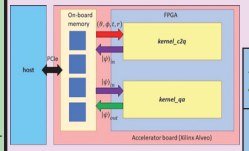
Background

- **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

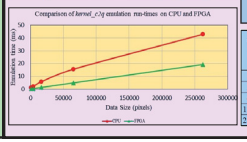


Classical-to-Quantum

- **Classical-to-quantum (C2Q)**
  - Synthesize arbitrary quantum state from ground state qubits
  - Uniformly-controlled operations ($\mathbf{U_j}$) + Hadamard (**H**) gates
  - Parameters for $\mathbf{U_j}$ 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

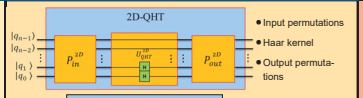


	Number of data items (N)	Number of qubits (n)	Simulation (MATLAB / ibm_qasm) Gate Count	Simulation (MATLAB / ibm_qasm) Circuit Depth	Implementation (ibmq_16_melbourne) Gate Count	Implementation (ibmq_16_melbourne) Circuit Depth
Complex randomized data	4.00E+00	2	7.00E+00	6.00E+00	1.70E+01	1.50E+01
	1.60E+01	4	3.30E+01	3.00E+01	9.20E+01	7.20E+01
	6.40E+01	6	1.31E+02	1.20E+02	3.38E+02	2.76E+02
	2.56E+02	8	5.17E+02	5.10E+02		
	1.02E+03	10	2.06E+03	2.05E+03	NA due to hardware limitations of IBM-Q	
	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		

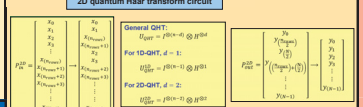


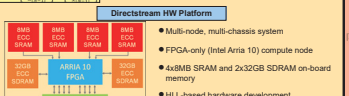
Data size (no. of pixels)	No. of states (N)	No. of qubits (n)	kernel_c2q emulation time (ms) CPU	kernel_c2q emulation time (ms) FPGA	kernel_qft emulation time (ms) CPU	kernel_qft emulation time (ms) FPGA	Total emulation time (ms) CPU	Total emulation time (ms) FPGA	Speedup
16x16x3	1024	10	0.98	0.081	0.464	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



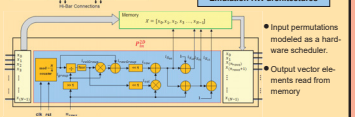
- Input permutations
- Haar kernel
- Output permutations



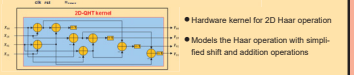


- Multi-node, multi-chassis system
- FPGA-only (Intel Arria 10) compute node
- 4x8MB SRAM and 2x32GB SDRAM on-board memory
- HLL-based hardware development

Emulation HW architectures



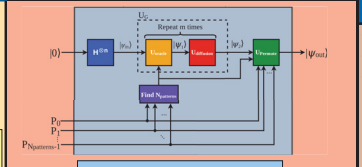
- Input permutations modeled as a hardware scheduler.
- Output vector elements read from memory

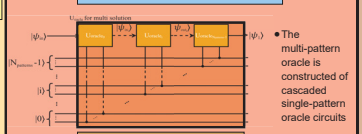


- Hardware kernel for 2D Haar operation
- Models the Haar operation with simplified shift and addition operations

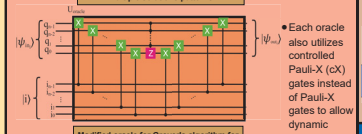
Number of qubits, n	Number of pixels	On-chip* resource utilization (%) ALMs	BRAMs	DSPs	QBM utilization (bytes) SDRAM	Emulation time (sec)
8	16x16	14	9	2	4K	6.73E-06
10	32x32	14	9	2	16K	1.66E-05
12	64x64	14	9	2	64K	5.62E-05
14	128x128	14	9	2	256K	0.0002
16	256x256	14	9	2	1M	0.0008
18	512x512	14	9	2	4M	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

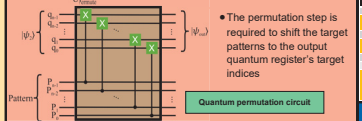




- The multi-pattern oracle is constructed of cascaded single-pattern oracle circuits



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

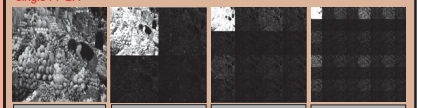


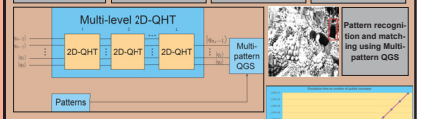
- The permutation step is required to shift the target patterns to the output quantum register's target indices

Emulation 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/ multi	32	32-bit floating pt.	233

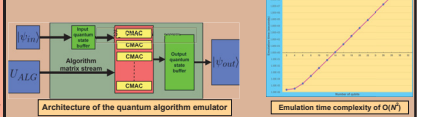
Dimension Reduction and Pattern Recognition and Matching

- High-resolution test images up to 64*k* × 64*k* 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





Pattern recognition and matching using Multi-pattern QGS



Architecture of the quantum algorithm emulator

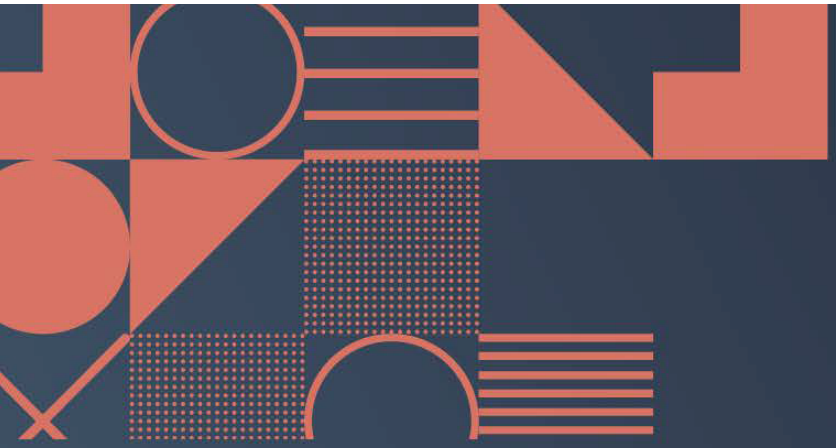
Emulation time complexity of $O(N^2)$

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			94E-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			13.82s
	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

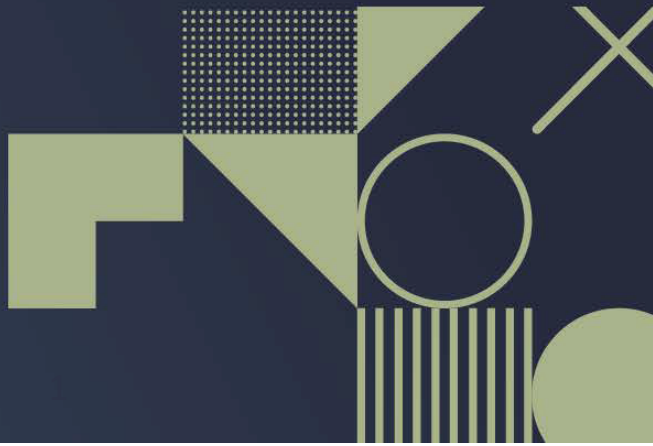


Towards Complete Emulation of Quantum Algorithms using High-Performance Reconfigurable Computing

Naveed Mahmud

Department of Electrical Engineering and Computer Science
University of Kansas

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



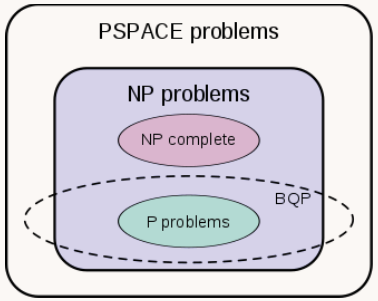
Outline

- Introduction and Motivations
- Background
- Proposed Work
- Experimental Results
- Conclusions

Introduction and Motivations

- Why Quantum Computing?
  - Solving specific problems (BQP)
  - Large multi-dimensional data space
  - Speedup over classical
- Challenges
  - NISQ → Decoherence noise
  - Running deep circuits
  - Low fidelity
  - Classical-to-quantum data encoding (Arbitrary State Synthesis)
  - Investigating I/O intensive applications

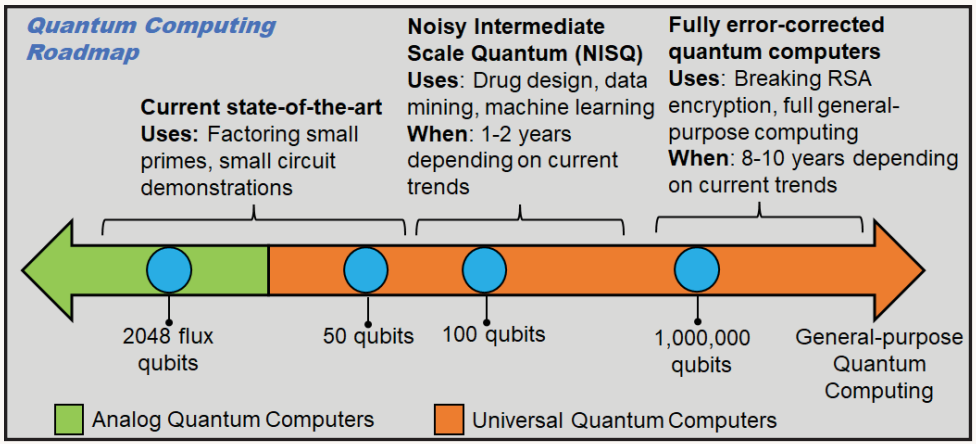
NISQ ≡ Noisy-Intermediate-Scale-Quantum



Bounded-error Quantum Polynomial (BQP): class of problems solvable by quantum computers in polynomial time.

PSPACE: Polynomial Space
NP: Non-deterministic Polynomial

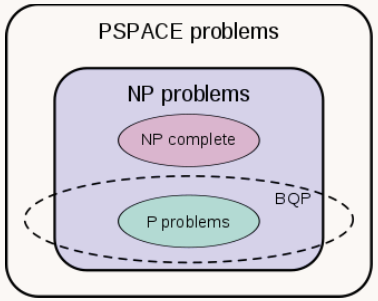
Source:
Nielsen, Michael; Chuang, Isaac (2000). Quantum Computation and Quantum Information. Cambridge: Cambridge University Press



Introduction and Motivations

- Why Quantum Computing?
  - Solving specific problems (BQP)
  - Large multi-dimensional data space
  - Speedup over classical
- Challenges
  - NISQ → Decoherence noise
  - Running deep circuits
  - Low fidelity
  - Classical-to-quantum data encoding (Arbitrary State Synthesis)
  - Investigating I/O intensive applications

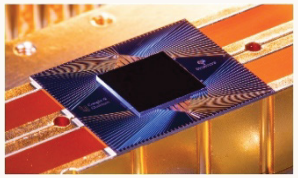
NISQ ≡ Noisy-Intermediate-Scale-Quantum



Bounded-error Quantum Polynomial (BQP): class of problems solvable by quantum computers in polynomial time.

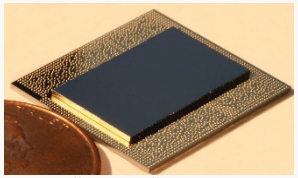
PSPACE: Polynomial Space
NP: Non-deterministic Polynomial

Source:
Nielsen, Michael; Chuang, Isaac (2000). Quantum Computation and Quantum Information. Cambridge: Cambridge University Press

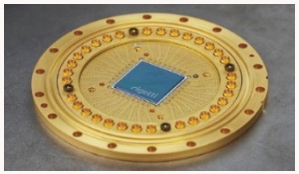


Google "Sycamore": 54 qubits, quantum volume: 32

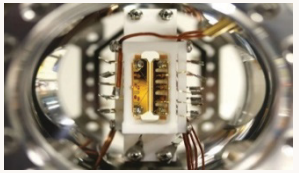
Honeywell: 10 qubits, quantum volume: 64



IBM-Q "Falcon": 27 qubits, quantum volume: 128



Rigetti "Aspen-9": 32 qubits, quantum volume: N/A

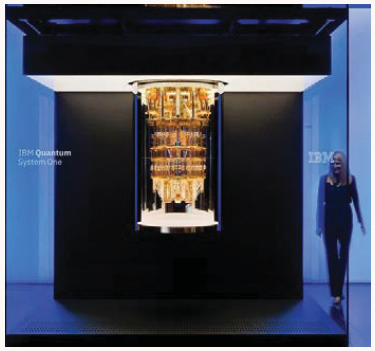


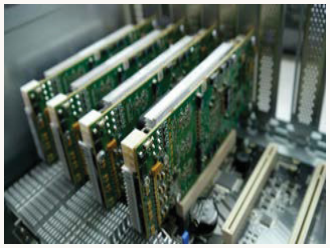
IonQ: 32 qubits, quantum volume: > 4E6

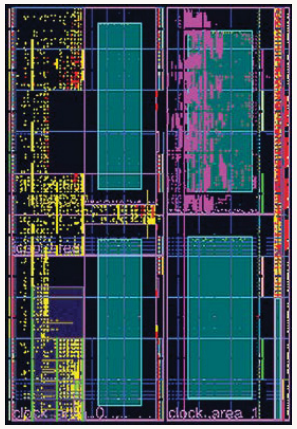
D-Wave "Advantage": > 5000 qubits, 15-way connectivity

Introduction and Motivations

- Need for Simulation / Emulation
  - Difficulty of reliable operation
  - High cost of access
    - E.g., academic hourly rate of $1,250 up to 499 annual hours
  - Verification and benchmarking
- Emulation using FPGAs
  - Cost-effective
  - Greater speedup vs. SW
  - Exploiting parallelism
  - Dynamic (reconfigurable) vs. fixed architectures
  - Limitations → Scalability, Accuracy



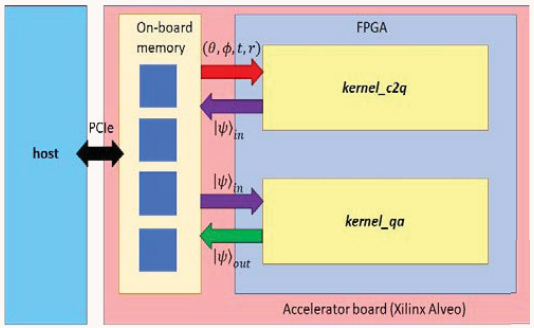


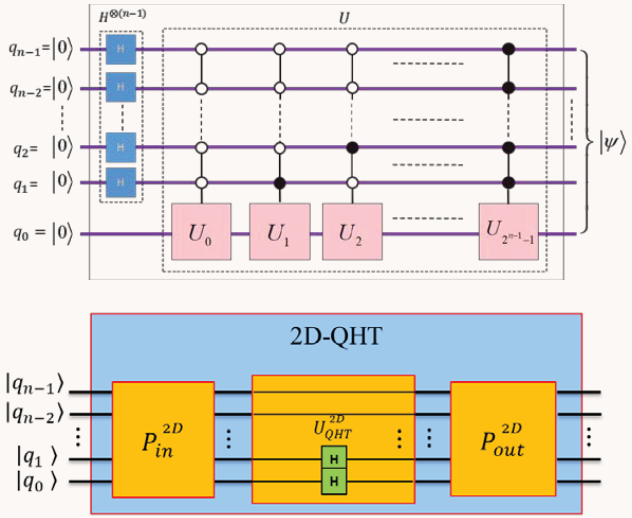


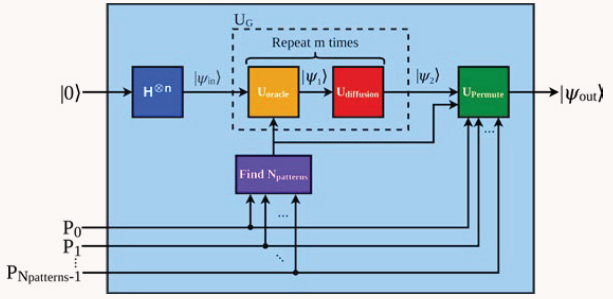
Introduction and Motivations

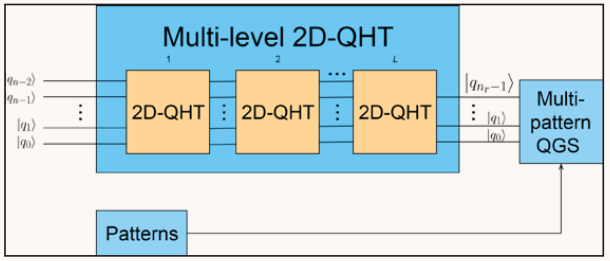
•Contributions

- •A cost-effective, High-Level Synthesis (HLS) and FPGA accelerated methodology for complete emulation of quantum algorithms
- •Emulating circuits for classical-to-quantum (C2Q) data encoding (arbitrary state synthesis) and various quantum algorithms
- •Dimension reduction for dynamic pattern recognition on high-resolution, spatio-spectral data







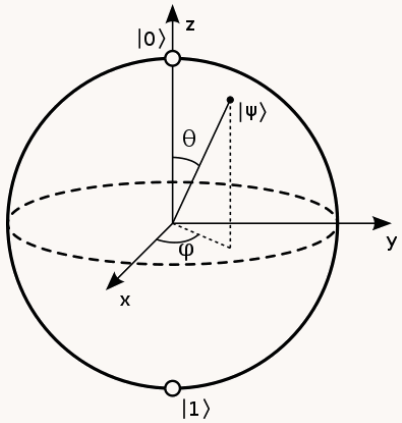


Outline

- Introduction and Motivations
- Background
- Proposed Work
- Experimental Results
- Conclusions

Background – Quantum Computing

- Qubits
  - Physical implementations
    - Electron (spin)
    - Nucleus (spin through NMR)
    - Josephson junction (superconducting qubits)
    - Trapped ions
  - Bloch sphere representation
    - Basis states → $|0\rangle$, $|1\rangle$
    - Pure states → $|\psi\rangle$
    - Local phase (azimuth) angle ϕ and local rotation (elevation) angle θ
    - Vector of complex coefficients
- Superposition
  - Linear sum of distinct basis states
  - Applies to state with *n*-qubits
- Entanglement
  - Entangled state cannot be factored into a tensor product
  - Measuring a qubit gives information about other qubits



Single-Qubit Superposition: $|\psi_1\rangle = \alpha|0\rangle + \beta|1\rangle \equiv \begin{bmatrix}\alpha\\ \beta\end{bmatrix}$

Born Rule: $p(|\psi_1\rangle \to |0\rangle) = |\alpha|^2, \; p(|\psi_1\rangle \to |1\rangle) = |\beta|^2$

Multi-Qubit Superposition:

$$|\psi_3\rangle = |q_2 q_1 q_0\rangle = |q_2\rangle \otimes |q_1\rangle \otimes |q_0\rangle = \begin{bmatrix}\alpha_2\\ \beta_2\end{bmatrix} \otimes \begin{bmatrix}\alpha_1\\ \beta_1\end{bmatrix} \otimes \begin{bmatrix}\alpha_0\\ \beta_0\end{bmatrix}$$

$$|\psi_3\rangle = \alpha_2\alpha_1\alpha_0|000\rangle + \alpha_2\alpha_1\beta_0|001\rangle + \ldots + \beta_2\beta_1\beta_0|111\rangle$$

$$|\psi_3\rangle = c_0|0\rangle + c_1|1\rangle + \ldots + c_7|7\rangle \Rightarrow |\psi_n\rangle = \sum_{q=0}^{2^n-1} c_q |q\rangle$$

Born Rule: $p(|\psi_n\rangle \to |q\rangle) = |c_q|^2 \Rightarrow \||\psi_n\rangle\|^2 = \sum_{q=0}^{2^n-1} |c_q|^2 = 1$

Multi-Qubit Entanglement:

$$\left(|\psi_n\rangle_{\text{entangled}} = |q_{n-1}\ldots q_1 q_0\rangle_{\text{entangled}}\right) \neq \left(|\psi_n\rangle_{\text{un-entangled}} = |q_{n-1}\rangle \otimes \ldots |q_1\rangle \otimes |q_0\rangle\right)$$

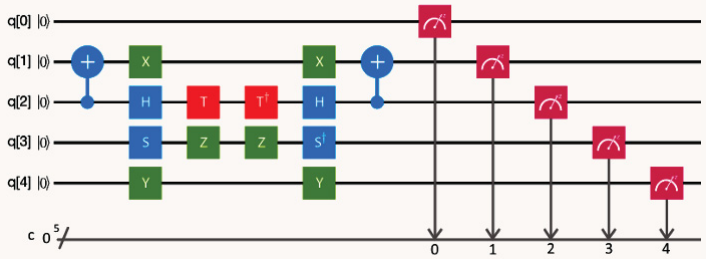
$$\textit{For Example}: \left(|\psi_2\rangle_{\text{entangled}} = |q_1 q_0\rangle_{\text{entangled}}\right) \neq \left(|q_1\rangle \otimes |q_0\rangle = \begin{bmatrix}\alpha_1\\ \beta_1\end{bmatrix} \otimes \begin{bmatrix}\alpha_0\\ \beta_0\end{bmatrix}\right)$$

$$|\psi_2\rangle_{\text{entangled}} = c_0|00\rangle + c_3|11\rangle \neq \alpha_1\alpha_0|00\rangle + \alpha_1\beta_0|01\rangle + \beta_1\alpha_0|10\rangle + \beta_1\beta_0|11\rangle$$

NMR ≡ **N**uclear **M**agnetic **R**esonance

Background – Circuit Model

- Quantum Gates
  - Hadamard Gate
  - Swap Gate
  - Controlled-NOT gate
  - Controlled Rotation Gates
- Quantum Circuit Depth
  - Total number of gates (levels) in the longest path, i.e., total number of time-steps
- Gate count
  - Total number of gates in circuit



Depth: 6
Gate count: 14

Gate	Symbol	Matrix
Hadamard	H	$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$
Pauli X	X	$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$
Swap		$\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}$
Controlled phase shift	R_ϕ	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&e^{i\phi}\end{bmatrix}$
Controlled NOT		$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$
Controlled Pauli Z	Z	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&-1\end{bmatrix}$

Gate	Symbol	Matrix
Controlled Rotation x	R_x	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&\cos\left(\frac{\alpha}{2}\right)&-i\sin\left(\frac{\alpha}{2}\right)\\0&0&-i\sin\left(\frac{\alpha}{2}\right)&\cos\left(\frac{\alpha}{2}\right)\end{bmatrix}$
Controlled Rotation y	R_y	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&\cos\left(\frac{\theta}{2}\right)&-\sin\left(\frac{\theta}{2}\right)\\0&0&\sin\left(\frac{\theta}{2}\right)&\cos\left(\frac{\theta}{2}\right)\end{bmatrix}$
Controlled Rotation z	R_z	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&e^{-i\frac{\phi}{2}}&0\\0&0&0&e^{i\frac{\phi}{2}}\end{bmatrix}$
Controlled U	U	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&U_{00}&U_{01}\\0&0&U_{10}&U_{11}\end{bmatrix}$

Outline

- Introduction and Motivations
- Background
- Proposed Work
- Experimental Results
- Conclusions

Arbitrary State Synthesis

- Formal Model
  - Classical data is normalized as N coefficients of a quantum state $|\psi\rangle$, $\boldsymbol{N = 2^n}$, where $\boldsymbol{n}$ is the number of qubits

$$|\psi\rangle = |q_{n-1}\rangle \otimes |q_{n-2}\rangle \otimes \dots \otimes \cdots |q_1\rangle \otimes |q_0\rangle$$
$$= |q_{n-1}q_{n-2}\dots q_1 q_0\rangle = \sum_{i=0}^{N-1} \alpha_i |i\rangle$$

  - Pauli decomposition of single qubit

$$|\psi\rangle = R_z(\phi) \cdot R_y(\theta) \cdot re^{i\frac{t}{2}} \cdot |0\rangle$$

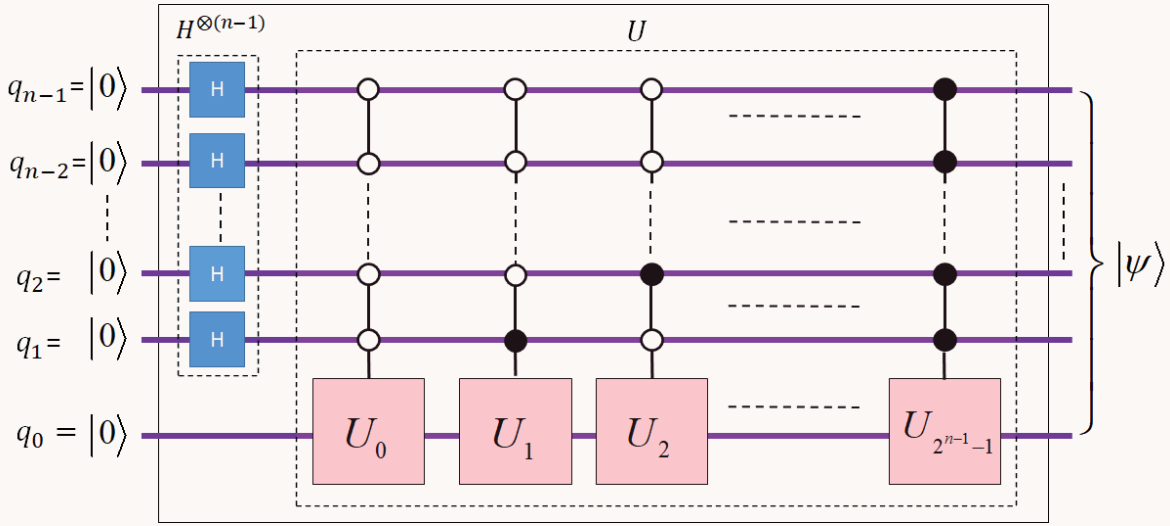
  - $(\boldsymbol{r_j, t_j, \theta_j, \varphi_j})$ parameters extracted from $\boldsymbol{j^{th}}$ coefficient pair $(\boldsymbol{\alpha, \beta})$, $\boldsymbol{j = 0, 1, 2, \dots, \frac{N}{2} - 1}$

$$|\alpha| = \sqrt{\mathrm{Re}^2(\alpha) + \mathrm{Im}^2(\alpha)}, \qquad \angle\alpha = \tan^{-1}\left(\frac{\mathrm{Im}(\alpha)}{\mathrm{Re}(\alpha)}\right)$$
$$|\beta| = \sqrt{\mathrm{Re}^2(\beta) + \mathrm{Im}^2(\beta)}, \qquad \angle\beta = \tan^{-1}\left(\frac{\mathrm{Im}(\beta)}{\mathrm{Re}(\beta)}\right)$$

$$r = \sqrt{|\alpha|^2 + |\beta|^2}\,, \qquad t = \angle\beta + \angle\alpha$$
$$\theta = 2\tan^{-1}\left(\frac{|\beta|}{|\alpha|}\right), \qquad \phi = \angle\beta - \angle\alpha$$

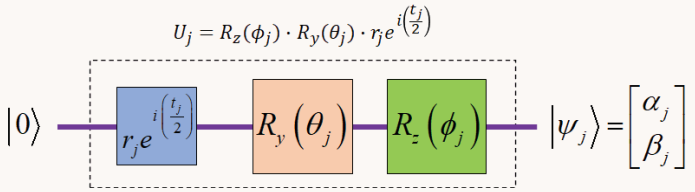
Arbitrary State Synthesis

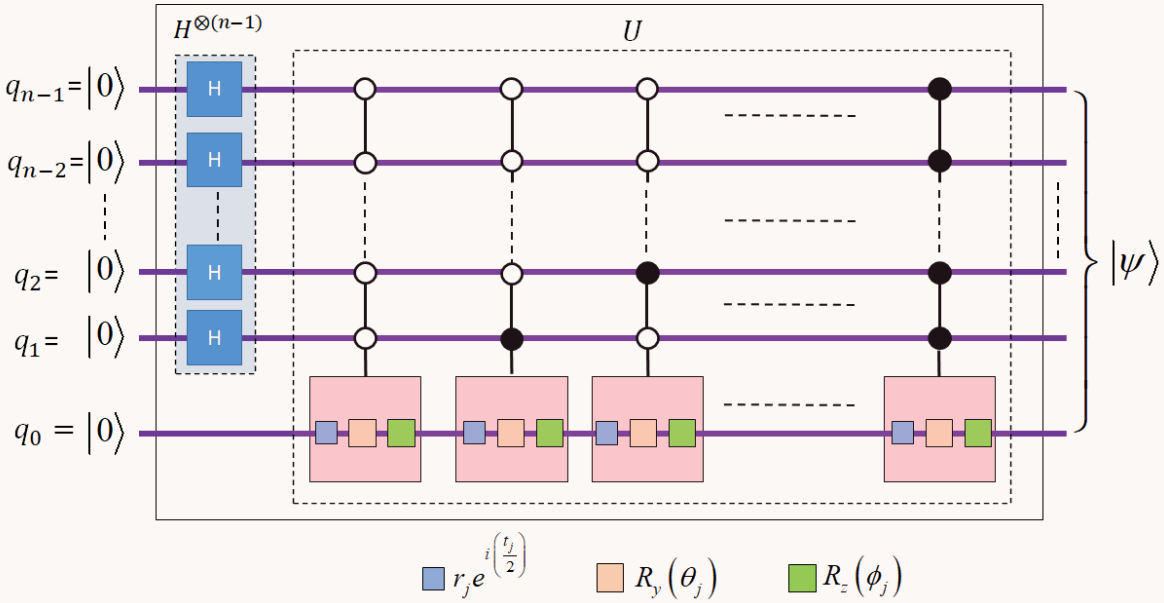
- Quantum Circuits for Classical-to-quantum
  - Parameters are used in a conditional logic circuit
    - Hadamard gates for equal superposition
    - Uniformly-controlled unitary operations $\boldsymbol{U_0}, \boldsymbol{U_1}, \ldots, \boldsymbol{U_j}, \ldots \boldsymbol{U_{2^{n-1}-1}}$



Arbitrary State Synthesis

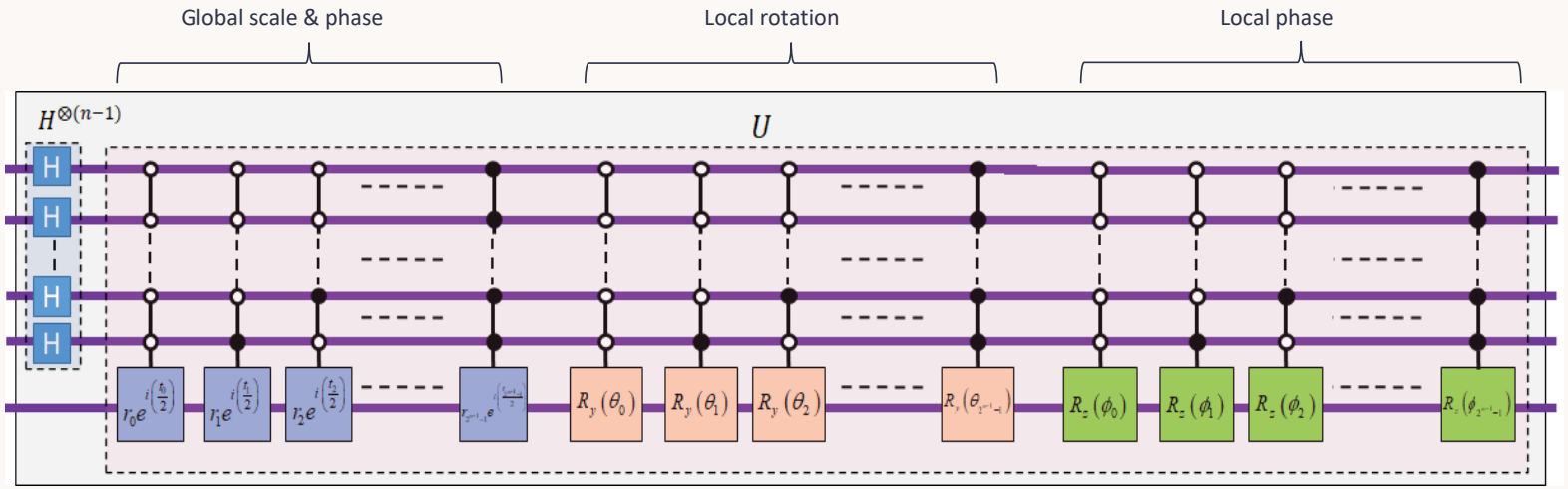
- Quantum Circuits for Classical-to-quantum
  - Parameters are used in a conditional logic circuit
    - Each U_j transformation can be factored by Pauli decomposition
    - Every j^{th} decomposition requires $(r_j, t_j, \theta_j, \varphi_j)$





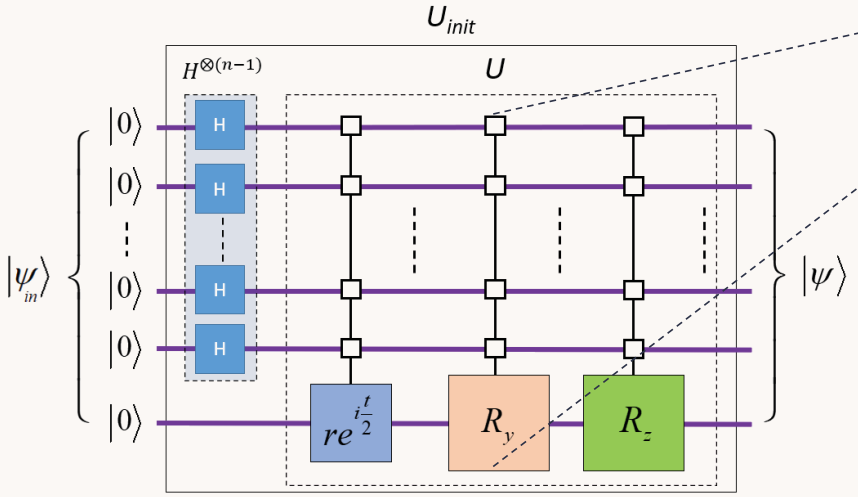
Arbitrary State Synthesis

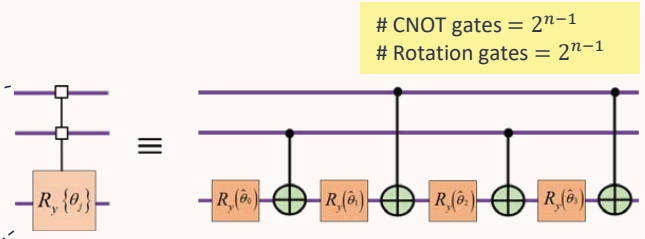
- Quantum Circuits for Classical-to-quantum
  - Each set of operations are mutually exclusive
  - Expansion into groups of uniformly-controlled *global scale & phase*, *local rotation*, and *local phase*



Arbitrary State Synthesis

- Quantum Circuits for Classical-to-quantum
  - Uniformly controlled operations expressed as single-gate operations (with square-box notation)
  - Final decomposition into primitive CNOT and single-qubit rotation gates





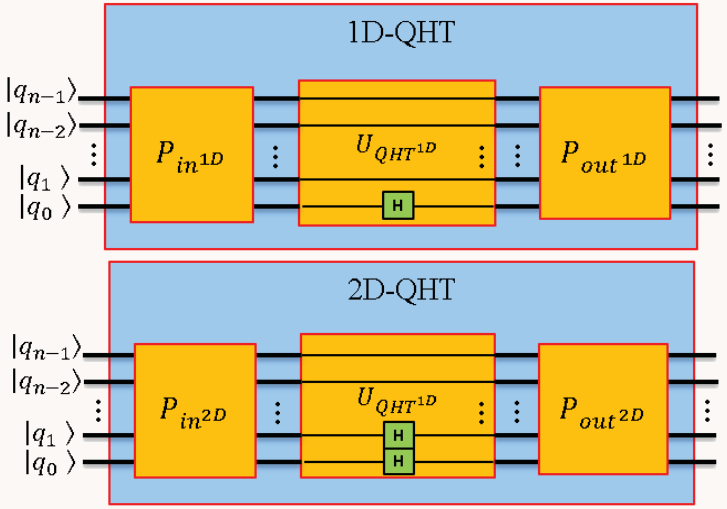
$n \equiv number\ of\ qubits$

Method	Gate Count	Circuit Depth
Mottonen [17], 2004	2^{n+2} - 6	2^{n+2} - 6
Shende [18], 2006	2^{n+2} - 2n	2^{n+2} - 2n
Niemann [19], 2016	2^{n+2} + 4n - 9	2^{n+2} + 3n - 8
Proposed	**2^{n+1} + n - 3**	**2^{n+1} - 2**

Comparison of proposed method to current methods

Quantum Haar Transform

- QHT operations
  - Haar wavelet function, U_{QHT}
  - Perfect shuffle permutations (PSP), P_{in}, P_{out}
- Haar wavelet function generalized
  - n qubits
  - d-dimensional kernel



Proposed circuits for 1D and 2D QHT

$$P_{in}^{2D} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{(n_{rows})} \\ x_{(n_{rows}+1)} \\ x_{(n_{rows}+2)} \\ x_{(n_{rows}+3)} \\ \vdots \\ \vdots \\ x_{(N-1)} \end{bmatrix} \rightarrow \begin{bmatrix} x_0 \\ x_1 \\ x_{(n_{rows})} \\ x_{(n_{rows}+1)} \\ x_2 \\ x_3 \\ x_{(n_{rows}+2)} \\ x_{(n_{rows}+3)} \\ \vdots \\ \vdots \\ \vdots \\ x_{(N-1)} \end{bmatrix}$$

General QHT:

$$U_{QHT} = I^{\otimes(n-d)} \otimes H^{\otimes d}$$

For 1D-QHT, $d = 1$:

$$U_{QHT}^{1D} = I^{\otimes(n-1)} \otimes H^{\otimes 1}$$

For 2D-QHT, $d = 2$:

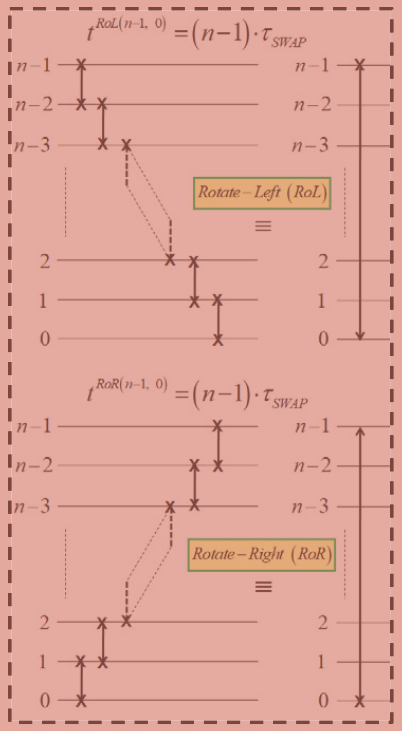
$$U_{QHT}^{2D} = I^{\otimes(n-2)} \otimes H^{\otimes 2}$$

$$P_{out}^{2D} = \begin{bmatrix} y_0 \\ y_{\left(\frac{n_{rows}}{2}\right)} \\ y_{\left(\frac{N}{2}\right)} \\ y_{\left(\left(\frac{n_{rows}}{2}\right)+\left(\frac{N}{2}\right)\right)} \\ \vdots \\ \vdots \\ y_{(N-1)} \end{bmatrix} \rightarrow \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ \vdots \\ y_{(N-1)} \end{bmatrix}$$

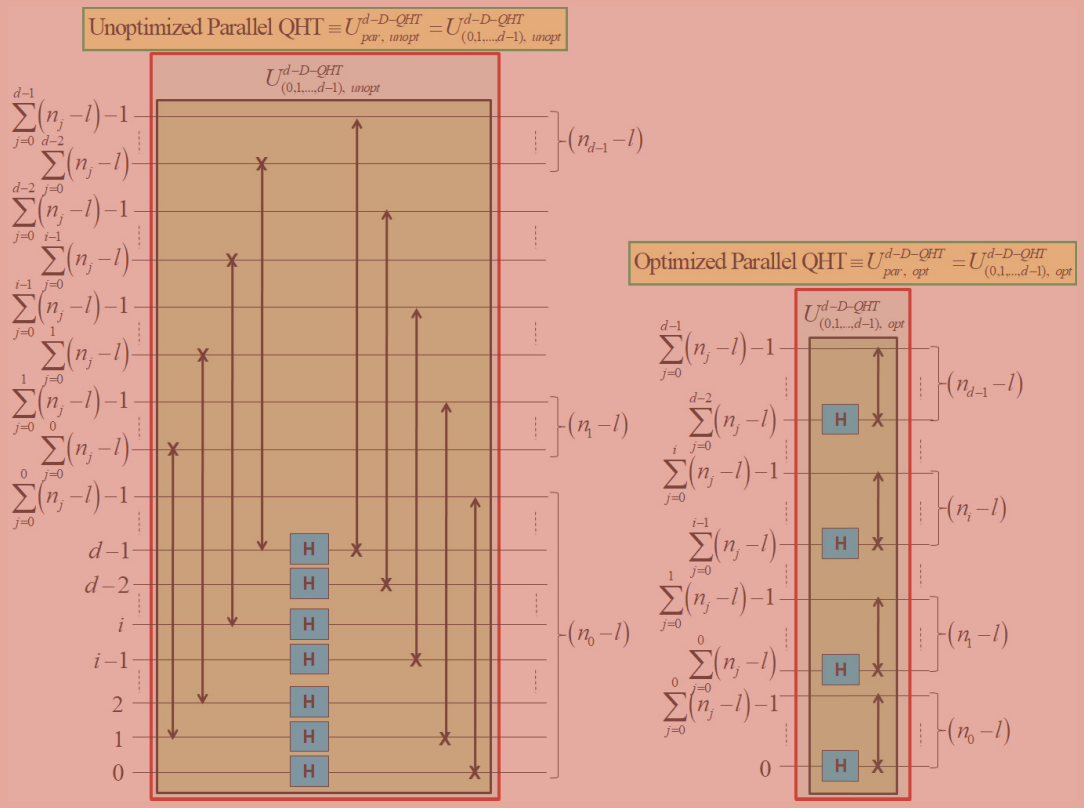
Quantum Haar Transform

Unpublished work

- QHT Circuit Optimization

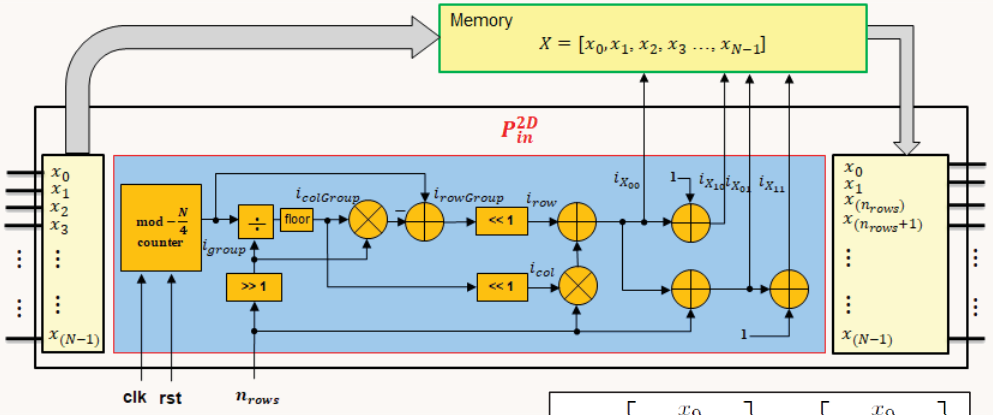


Rotate left and Rotate right operations

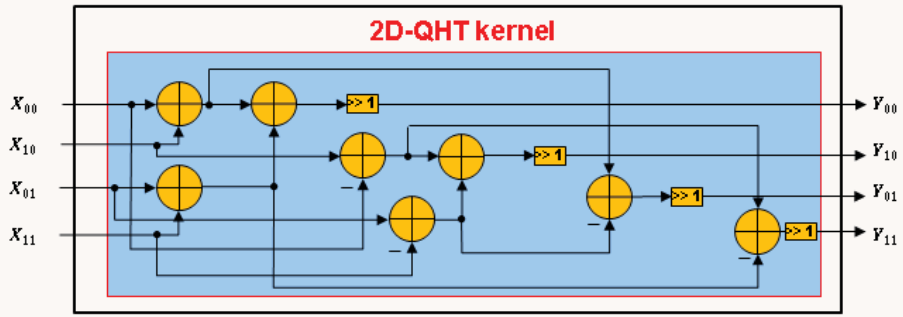


Quantum Haar Transform

- HW Architectures for QHT Emulation
  - Input permutations scheduler for P_{in}^{2D}
  - Haar operation kernel
  - For 2D-QHT, the kernel window size is 4



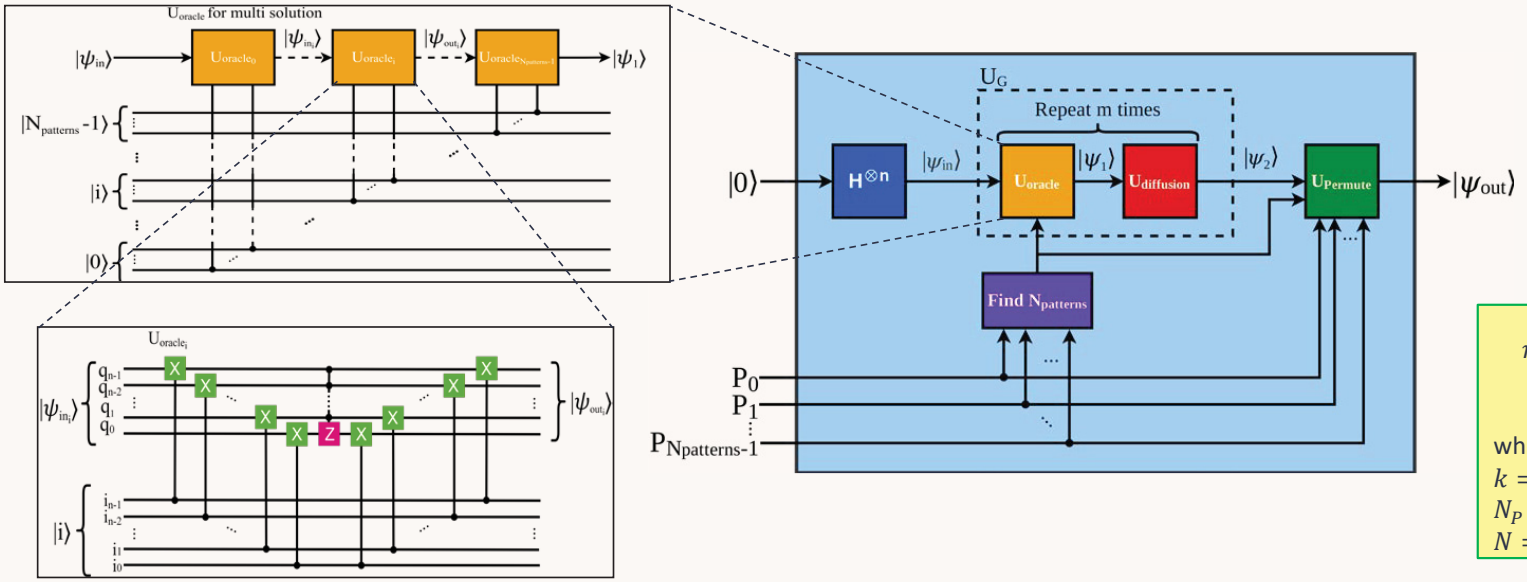
$$
\begin{aligned}
&// \text{ 2D-QHT kernel}\\
&Y_{00} = (X_{00} + X_{01} + X_{10} + X_{11})/2\\
&Y_{01} = (X_{00} - X_{01} + X_{10} - X_{11})/2\\
&Y_{10} = (X_{00} + X_{01} - X_{10} - X_{11})/2\\
&Y_{11} = (X_{00} - X_{01} - X_{10} + X_{11})/2
\end{aligned}
$$



$$
P_{\text{in}}^{2D}: \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{(n_{rows})} \\ x_{(n_{rows}+1)} \\ x_{(n_{rows}+2)} \\ x_{(n_{rows}+3)} \\ \vdots \\ x_{(N-1)} \end{bmatrix} \longmapsto \begin{bmatrix} x_0 \\ x_1 \\ x_{(n_{rows})} \\ x_{(n_{rows}+1)} \\ x_2 \\ x_3 \\ x_{(n_{rows}+2)} \\ x_{(n_{rows}+3)} \\ \vdots \\ x_{(N-1)} \end{bmatrix}
$$

Quantum Grover's Search

- Quantum Circuits for Dynamic Multi-pattern Quantum Grover's search (QGS)
  - The multi-pattern oracle is constructed of cascaded single-pattern oracle circuits
  - Each single-pattern oracle utilizes controlled Pauli-X (cX) gates instead of Pauli-X gates to allow dynamic search patterns

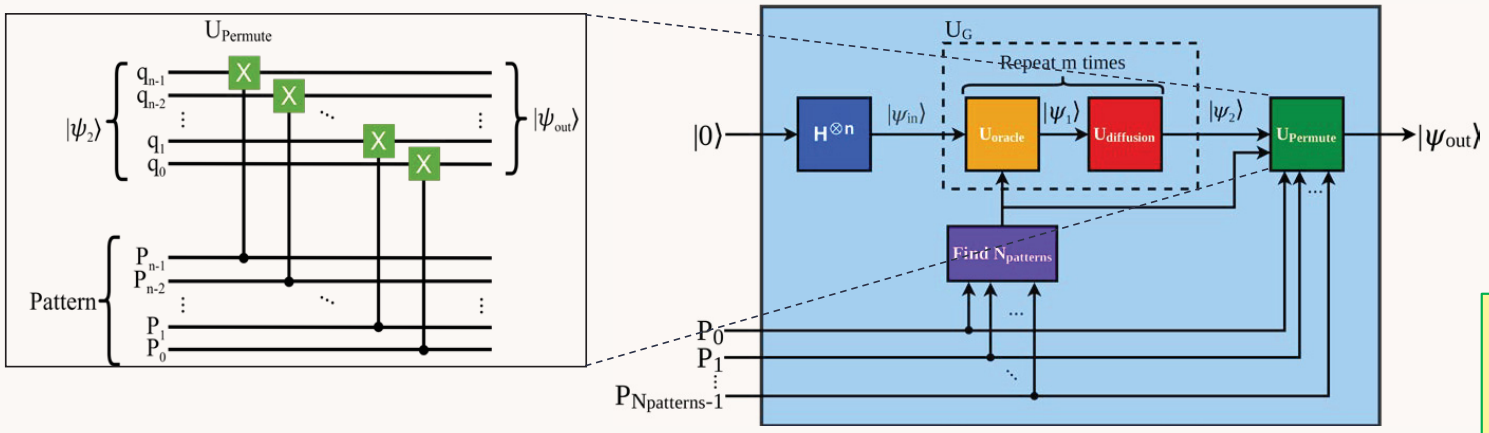


$$m = \left\lfloor \frac{\pi . k}{4 sin^{-1}\left(\sqrt{\frac{N_P}{N}}\right)} \right\rfloor$$

where,
$k = 1, 3, 5, 7, \ldots$
N_P = no. of patterns
N = no. of basis states

Quantum Grover's Search

- Quantum circuits for Dynamic Multi-pattern Quantum Grover's search (QGS)
  - The permutation step is required to shift the target patterns to the output quantum register's target indices

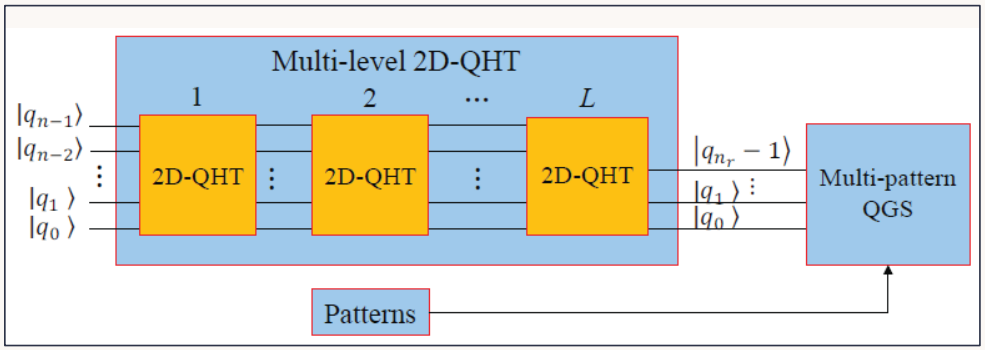


$$m = \left\lfloor \frac{\pi . k}{4sin^{-1}\left(\sqrt{\frac{N_P}{N}}\right)} \right\rfloor$$

where,
$k = 1, 3, 5, 7, \ldots$
N_P = no. of patterns
N = no. of basis states

Dimension Reduction and Pattern Matching

- Proposed Methodology
  - Dimension reduction using $\boldsymbol{L}$ levels of 2D quantum Haar transform (2D-QHT)
  - Single/multi-pattern searching using Quantum Grover's search (QGS)



Proposed system overview

$$L = \left\lfloor \frac{1}{2} log_2 \frac{N}{N_r} \right\rfloor$$

where,
N = no. of basis states (input image)
N_r = no. of basis states (reduced image)
and,
$N = 2^n, n$ = no. of qubits (input image)
$N_r = 2^{n_r}, n_r$ = no. of qubits (reduced image)

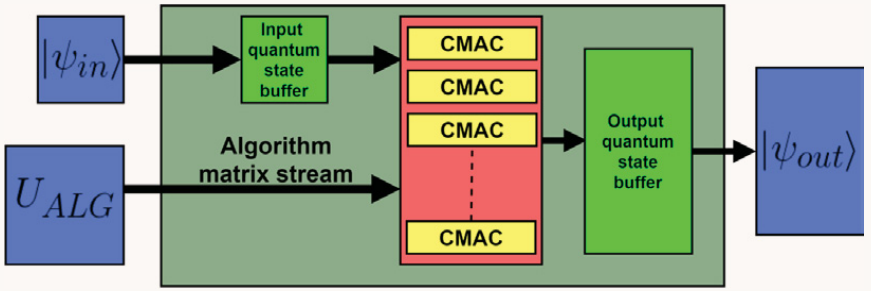
$$m = \left\lfloor \frac{\pi . k}{4sin^{-1}\left(\sqrt{\frac{N_P}{N}}\right)} \right\rfloor$$

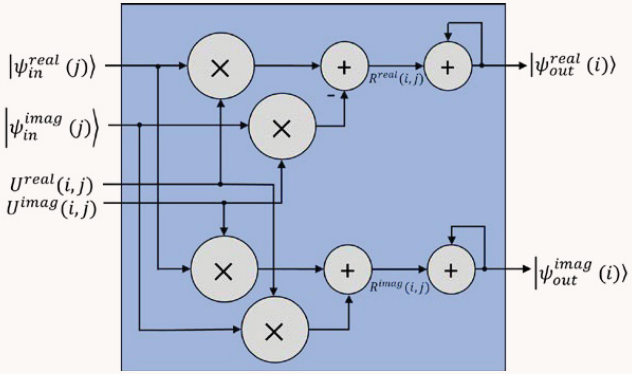
where
$k = 1, 3, 5, 7, \ldots$
N_P = no. of patterns
N = no. of basis states

Dimension Reduction and Pattern Matching

- HW Architecture for Emulation
  - Vector-matrix multiplication
    - Complex multiply-and-accumulate **(CMAC)**
  - Input quantum state vector, $|\psi_{in}\rangle$
  - Algorithm reduced to a unitary matrix, U_{ALG}
    - lookup / dynamic generation / stream
  - Output quantum state vector, $|\psi_{out}\rangle$

- Advantages
  - Generalized approach for any quantum algorithm
  - Independent of circuit depth
  - Low resource utilization and latency
  - High scalability
  - Parallelizable hardware architectures





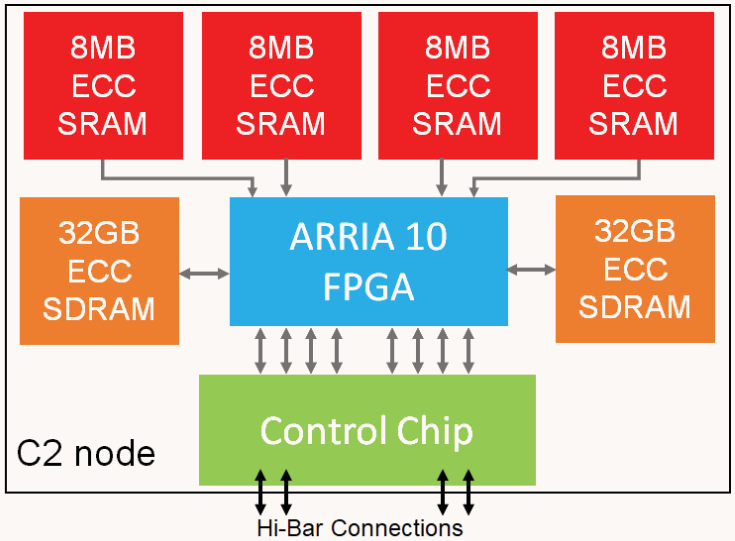
CMAC Architecture

Outline

- Introduction and Motivations
- Background
- Proposed Work
- Experimental Results
- Conclusions

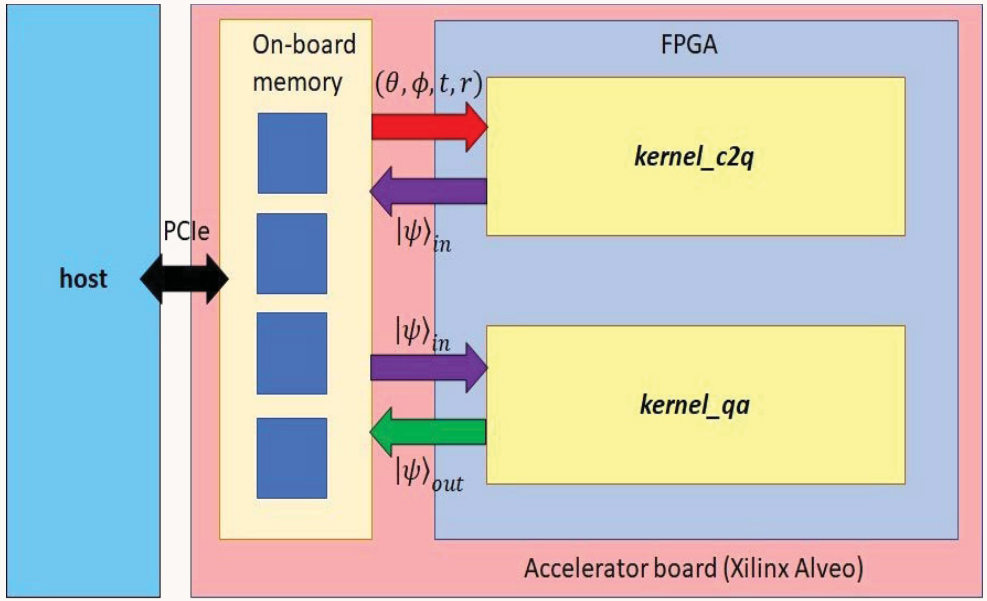
Experimental Setup

- Testbed Platform 1
  - High-performance reconfigurable computing (HPRC) system from DirectStream
  - OS-less, FPGA-only (Arria 10) architecture
  - Single node on-chip resources (OCR)
    - 427,200 Adaptive Logic Modules (ALMs)
    - 1,518 Digital signal Processors (DSPs)
    - 2,713 Block RAMs (BRAMs)
  - Single node on-board memory (OBM)
    - 2×32 GB SDRAM modules
    - 4×8 MB SRAM modules
  - Highly productive development environment
    - Parallel High-Level Language
    - C++-to-HW (previously Carte-C) compiler
    - Quartus Prime 17.0.2



Experimental Setup

- Testbed Platform 2
  - 16-core, 3 GHz AMD EPYC CPU with 251 GB system memory
  - Xilinx Alveo accelerator board with xcu250 FPGA, 4x16 GB on-board memory
  - Vitis unified software platform
    - OpenCL
    - C/C++ high-level synthesis
    - Register-transfer level (RTL)
  - Kernel partitions for classical-to-quantum (*kernel_c2q*) and quantum algorithm (*kernel_qa*)



Results – Arbitrary State Synthesis

- Simulation and implementation of proposed synthesis circuits
  - MATLAB
    - Noise-free qubit simulation
    - Up to 14-qubit circuits
  - IBM-Q
    - *qasm* simulator
    - *ibmq_16_melbourne* quantum processor
    - HW limitations
  - Input data
    - Complex randomized data
    - Real grayscale images
  - Measurements
    - Gate count
    - Circuit depth

	Number of data items (N)	Number of qubits (n)	Simulation (MATLAB / ibm_qasm)		Implementation (ibmq_16_melbourne)	
			Gate Count	Circuit Depth	Gate Count	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		

Results – Arbitrary State Synthesis

- Image reconstruction from synthesized quantum states
  - Fidelity (*F*) used as metric for similarity between expected state ($\psi_{expected}$) and measured state ($\psi_{measured}$)

$$F = \left|\langle\psi_{expected}|\psi_{measured}\rangle\right|^2$$

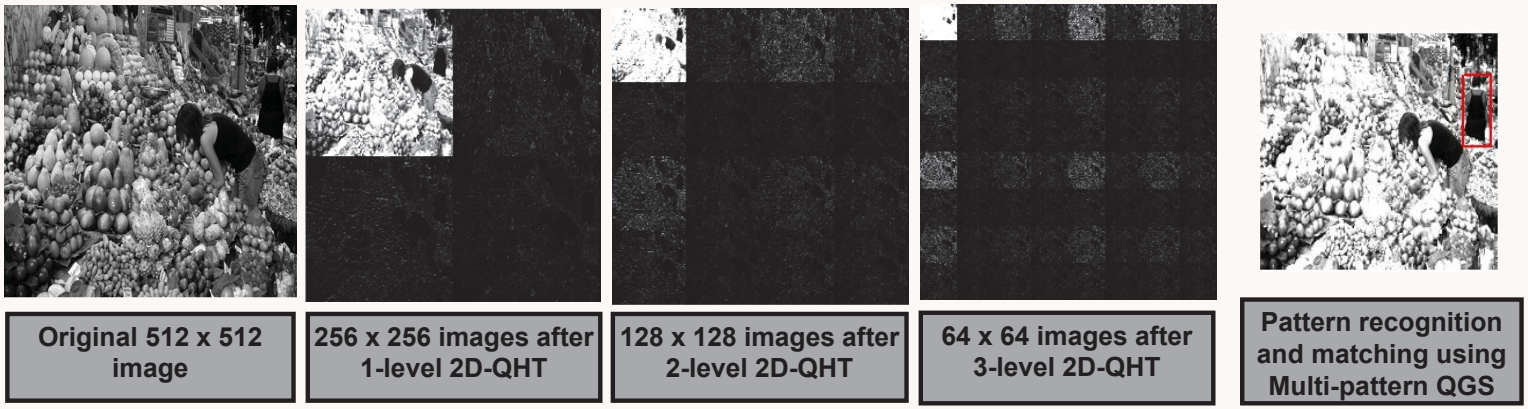
  - 16x16, 32x32, and 64x64 images encoded using 8-qubit, 10-qubit, and 12-qubit circuits
  - Partial corruption due to statistical noise in the NISQ device

Original image	Reconstructed image from simulations	
	MATLAB (noise-free)	IBM-QASM (NISQ† devices)
 16 x 16 pixels (8 qubits)		
	Fidelity = 100 %	Fidelity = 99.1644 %
 32 x 32 pixels (10 qubits)		
	Fidelity = 100 %	Fidelity = 96.5429 %
 64 x 64 pixels (12 qubits)		
	Fidelity = 100 %	Fidelity = 94.0894 %

†NISQ ≡ Noisy Intermediate-Scale Quantum

Results – Dimension Reduction and Pattern Matching

- Test Methodology
  - High-resolution test images up to 64k × 64k pixels
  - Multi-level packet 2D-QHT → L-level decomposition to target resolution
  - QGS returns indices of target pixels



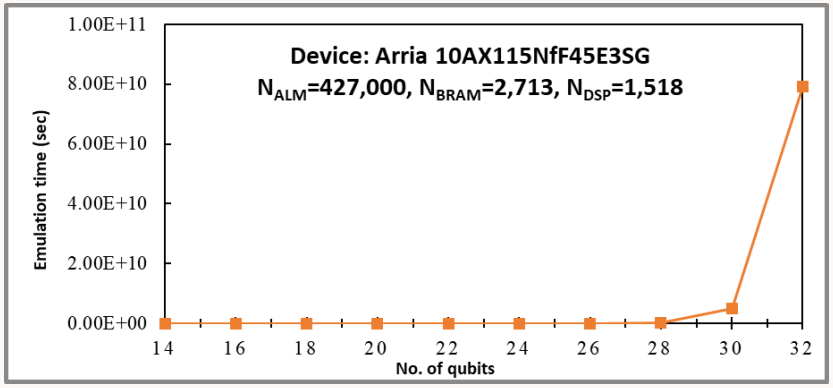
Original 512 x 512 image	256 x 256 images after 1-level 2D-QHT	128 x 128 images after 2-level 2D-QHT	64 x 64 images after 3-level 2D-QHT	Pattern recognition and matching using Multi-pattern QGS

Results – Dimension Reduction and Pattern Matching

- Combined QHT and QGS pattern recognition system implementation
  - Up to 32-qubit QHT and 10-qubit QGS circuits
  - $\mathbf{2} \times \mathbf{32GB}$ SDRAM OBM banks used to store input/output state vectors

No. of pixels	No. of qubits	No. of levels	OCR* utilization (%)			OBM** (bytes)	Emulation time (sec)***
			ALMs	BRAMs	DSPs		
128x128	14	3	22	16	2	128K	1.15E0
256x256	16	4	22	16	2	512K	1.84E01
512x512	18	5	22	16	2	2M	2.95E02
1024x1024	20	6	22	16	2	8M	4.72E03
2048x2048	22	7	22	16	2	32M	7.5E04
4096x4096	24	8	22	16	2	128M	1.2E06
8192x8192	26	9	22	16	2	512M	1.93E07
16Kx16K	28	10	22	16	2	2G	3.09E08
32Kx32K	30	11	22	16	2	8G	4.95E09
64Kx64K	32	12	22	16	2	32G	7.92E10

*Total on-chip resources: N_{ALM} = 427,000, N_{BRAM} = 2,713, N_{DSP} = 1,518
**Total on-board memory: 2x32 GB SDRAM banks
***Operating frequency: 233 MHz



System emulation time with number of qubits

ALM ≡ **A**daptive **L**ogic **M**odules
BRAM ≡ **B**lock **R**andom **A**ccess **M**emory
DSP ≡ **D**igital **S**ignal **P**rocessing block

Comparative Study

- Comparison with related work (FPGA-based emulation)

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 Dlugopolski (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**

Conclusions

- Near-future advantage of Quantum Computing
- Need for Quantum Emulation
  - FPGA-accelerated, cost-effective methodologies for complete emulation
- Proposed Methods
  - Depth optimized methods for arbitrary state synthesis
  - HLS-based emulation methodology
- Case studies
  - Multi-dimensional, multi-level quantum Haar Transform (QHT)
  - Single-pattern / multi-pattern quantum Grover's search (QGS)
  - Combining QHT + QGS for Dimension Reduction and pattern matching
- Experimental work
  - State-of-the-art test platforms from DirectStream, Xilinx, and IBMQ
  - Higher scalability, throughput, and accuracy compared to existing emulators
  - High fidelity of proposed quantum circuits


SC21
St. Louis, MO
science & beyond.
Thank you for listening!