Quantum Phase Estimation using Multivalued Logic

Agenda

- Importance of Quantum Phase Estimation (QPE)
- QPE using binary logic
- QPE using MVL
- Performance Requirements
- Salient features
- Conclusion

Introduction

- QPE – one of the most important quantum subroutines, used in :
  - 1) Shor’s Algorithm
  - 2) Abrams and Lloyd Algorithm
    - (Simulating quantum systems)
      - Calculation of Molecular Ground State Energies
  - 3) Quantum Counting (for Grover Search)
  - 4) Fourier Transform on arbitrary Z_p

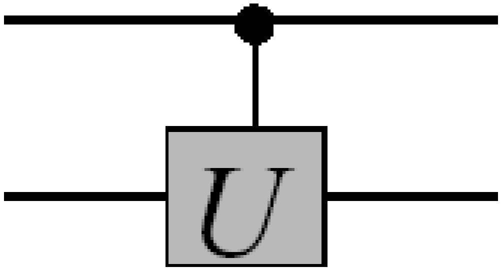
Abstract

- We generalize the *Quantum Phase Estimation algorithm* to MVL logic.
- We show the quantum circuits for QPE using **qudits**.
- We derive the performance requirements of the QPE to achieve high probability of success.
- We show how this leads to logarithmic decrease in the number of qudits and exponential decrease in error probability of the QPE algorithm as the value of the radix *d* increases.

General Controlled Gates

Controlled-U gate

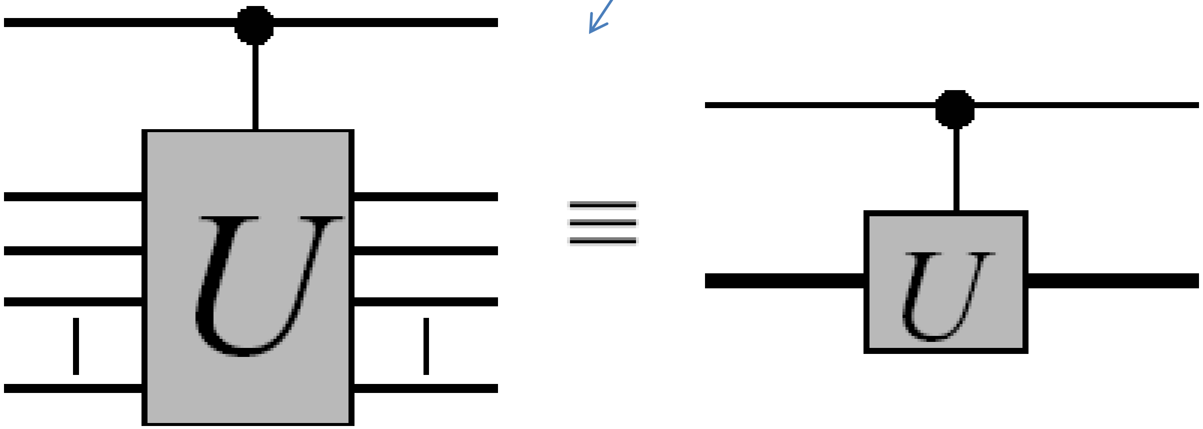
- Two-qubit controlled-U



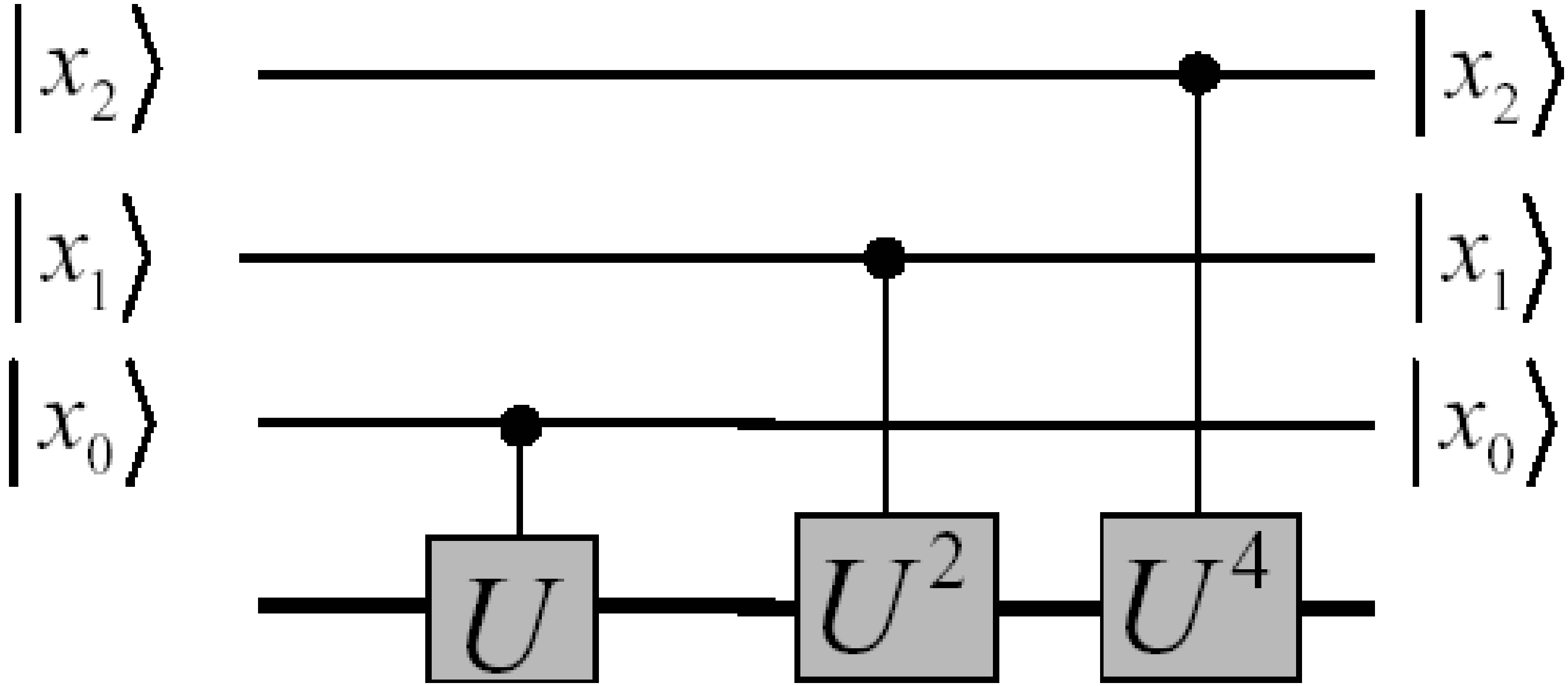
This is nothing new, CNOT, CV, CV+

- Multi-qubit controlled-U

This is a new concept, but essentially the same concept and mathematics



Controlled-U gate

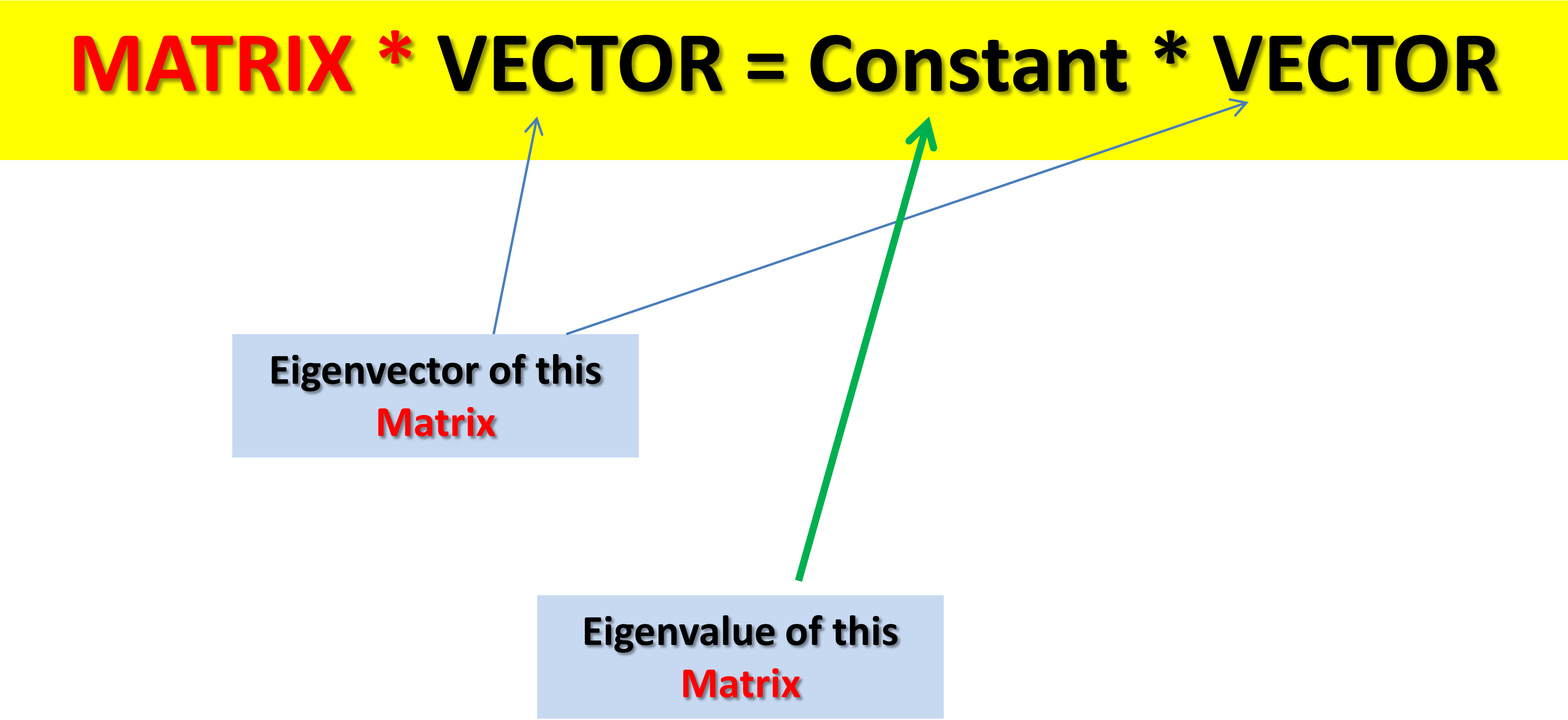


Another new concept – we use controlled powers of some Unitary Matrix U

$$U^{x_2x_1x_0}$$

REMINDER OF EIGENVALUES AND EIGENVECTORS

What is eigenvalue?



MATRIX * VECTOR = Constant * VECTOR

Eigenvector of this Matrix

Eigenvalue of this Matrix

Basic Math for Binary Phase Estimation

Phase estimation algorithm

- Given a unitary operator and an eigenstate of the operator
- The goal of the PE algorithm is to find the corresponding eigenvalue

Phase

$$\hat{U}\,|\phi\rangle = e^{i\phi}|\phi\rangle$$

Finding the eigenvalue is the same as finding its phase ϕ

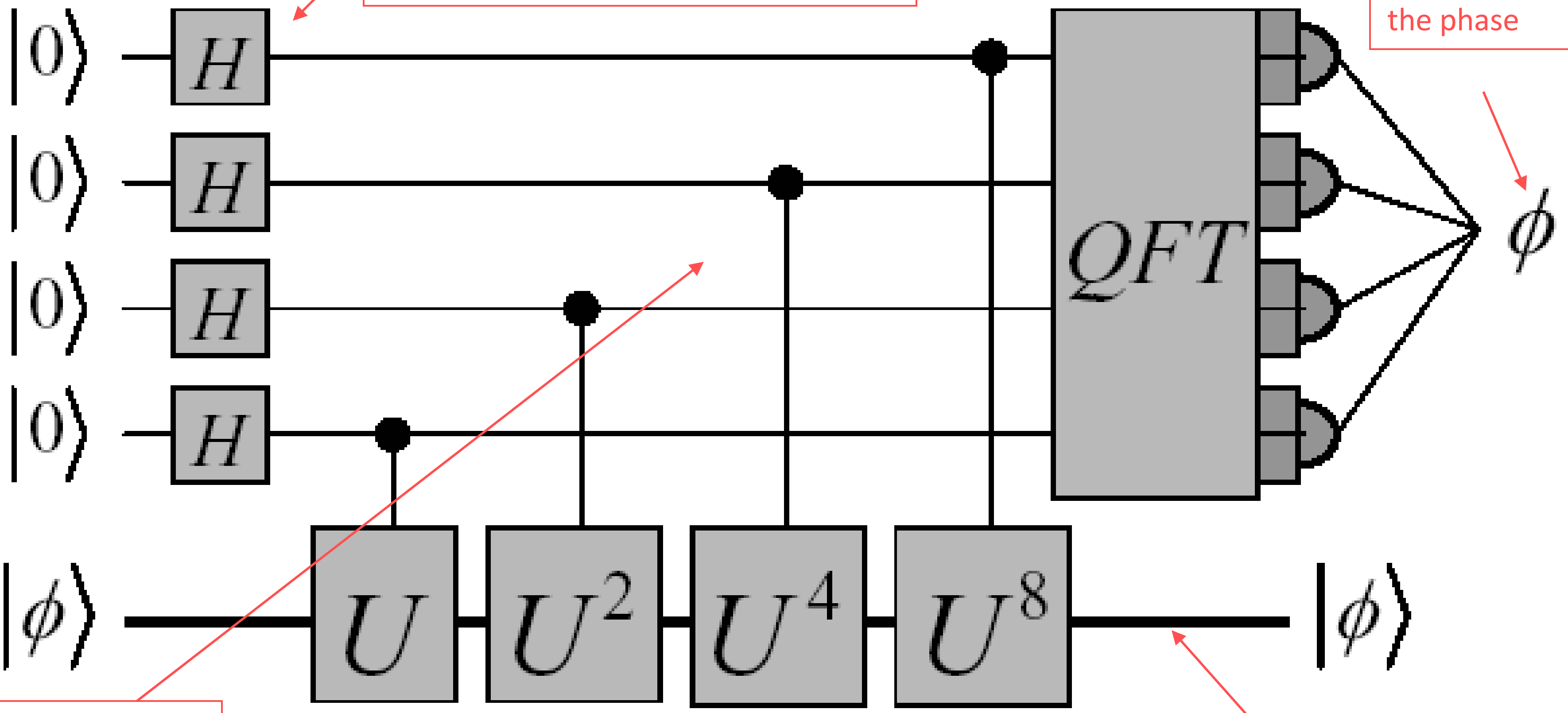
Phase estimation algorithm

- The PE algorithm uses two registers of qubits
  - The target register, to which U can be applied
  - The index register, which will be used to store the eigenvalue of U

Phase estimation algorithm

We initialize to all states

we measure the phase



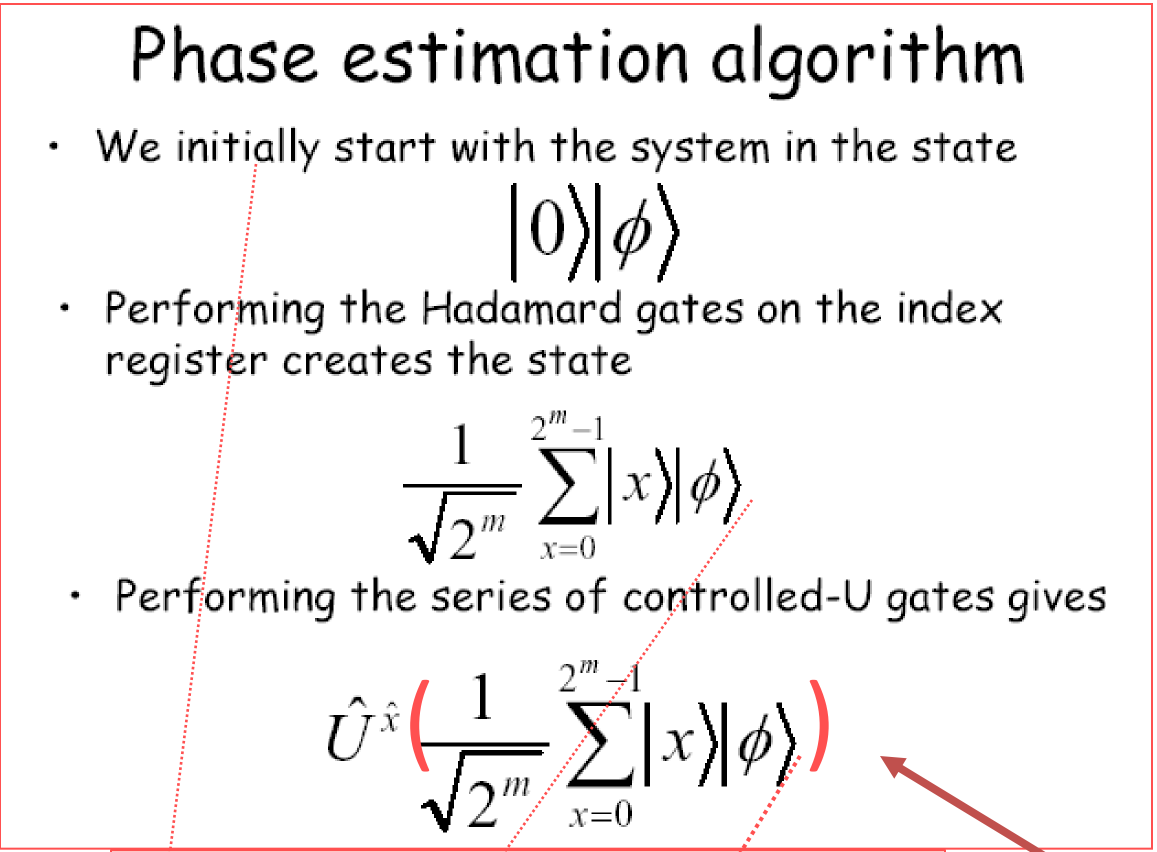
Index register to store the eigenvalue

Target register

Quantum circuit diagram

Unitary operator for which we calculate phase of eigenvalue using phase kickback

Formulas for the phase estimation algorithm



Phase estimation algorithm

- We initially start with the system in the state

$$|0\rangle|\phi\rangle$$

- Performing the Hadamard gates on the index register creates the state

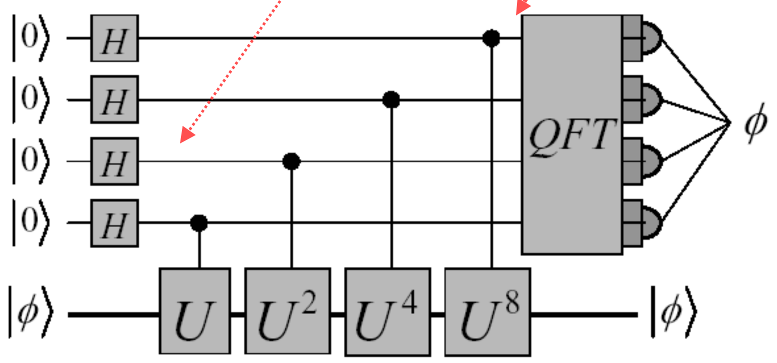
$$\frac{1}{\sqrt{2^m}}\sum_{x=0}^{2^m-1}|x\rangle|\phi\rangle$$

- Performing the series of controlled-U gates gives

$$\hat{U}^{\hat{x}}\left(\frac{1}{\sqrt{2^m}}\sum_{x=0}^{2^m-1}|x\rangle|\phi\rangle\right)$$

This is the input to QFT

Phase estimation algorithm



Quantum circuit diagram

Phase estimation algorithm

- We can move the U inside the summation

$$\frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle \hat{U}^x |\phi\rangle$$

- And replace U with $e^{i\phi}$

Because $e^{j\phi}$ is an eigenvallue of U

$$\frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle e^{ix\varphi} |\phi\rangle$$

Phase estimation algorithm

- Rearranging,

Assume k an integer

$$|\phi\rangle \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} e^{ix\varphi} |x\rangle \quad \text{if} \quad \phi = \frac{2\pi k}{2^m}$$

Number of bits

then

$$|\phi\rangle \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} e^{\frac{2\pi i x k}{2^m}} |x\rangle$$

Applying the quantum Fourier transform gives

$$|\phi\rangle |k\rangle$$

We found phase

Phase estimation algorithm

- Generally, k will not be an integer
- With high probability we will obtain the nearest integer to k
- Thus, we have an m-bit approximation to ϕ.

Concluding, we can calculate phase

Towards Generalization of Phase Estimation

Agenda

- Importance of Quantum Phase Estimation (QPE)
- QPE using binary logic
- QPE using MVL
- Performance Requirements
- Salient features
- Conclusion

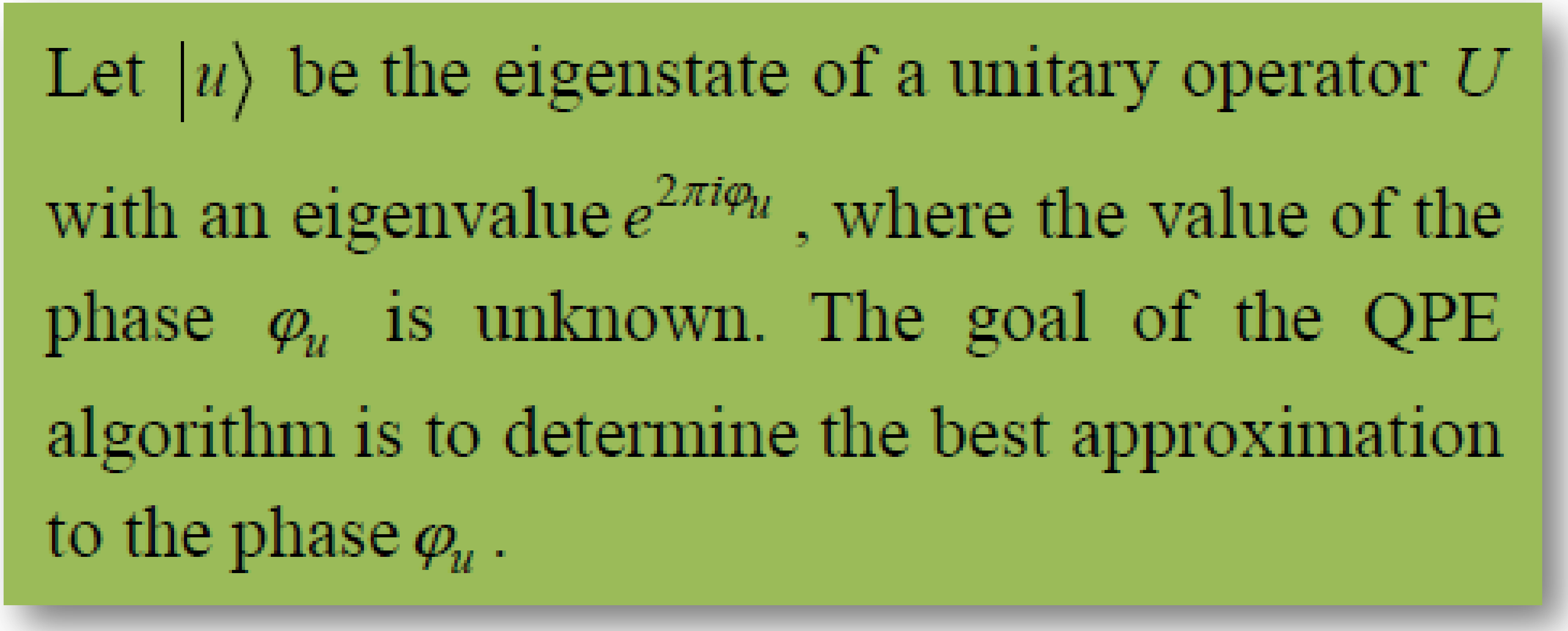
Abstract

- We generalize the *Quantum Phase Estimation algorithm* to MVL logic.
- We show the quantum circuits for QPE using **qudits**.
- We derive the performance requirements of the QPE to achieve high probability of success.
- We show how this leads to logarithmic decrease in the number of qudits and exponential decrease in error probability of the QPE algorithm as the value of the radix *d* increases.

Introduction

- QPE – one of the most important quantum subroutines, used in :
  - 1) Shor's Algorithm
  - 2) Abrams and Lloyd Algorithm
    - (Simulating quantum systems)
      - Calculation of Molecular Ground State Energies
  - 3) Quantum Counting (for Grover Search)
  - 4) Fourier Transform on arbitrary Z_p

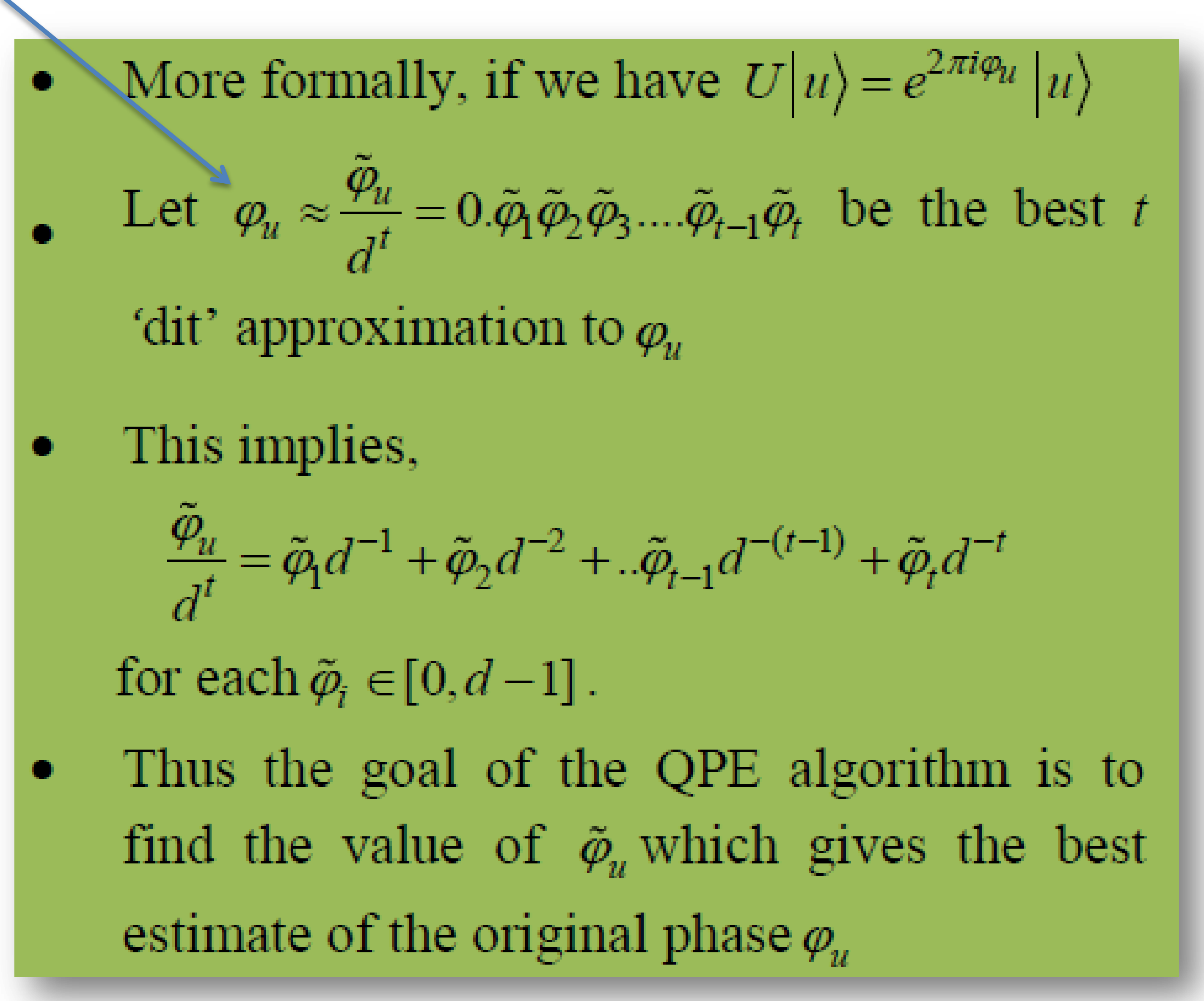
Quantum phase estimation (QPE)



Let $|u\rangle$ be the eigenstate of a unitary operator U with an eigenvalue $e^{2\pi i\varphi_u}$, where the value of the phase φ_u is unknown. The goal of the QPE algorithm is to determine the best approximation to the phase φ_u .

QPE – Formal Definition

phase



- More formally, if we have $U|u\rangle = e^{2\pi i \varphi_u}|u\rangle$
- Let $\varphi_u \approx \frac{\tilde{\varphi}_u}{d^t} = 0.\tilde{\varphi}_1\tilde{\varphi}_2\tilde{\varphi}_3....\tilde{\varphi}_{t-1}\tilde{\varphi}_t$ be the best t 'dit' approximation to φ_u
- This implies,

$$\frac{\tilde{\varphi}_u}{d^t} = \tilde{\varphi}_1 d^{-1} + \tilde{\varphi}_2 d^{-2} + ..\tilde{\varphi}_{t-1} d^{-(t-1)} + \tilde{\varphi}_t d^{-t}$$

for each $\tilde{\varphi}_i \in [0, d-1]$.

- Thus the goal of the QPE algorithm is to find the value of $\tilde{\varphi}_u$ which gives the best estimate of the original phase φ_u

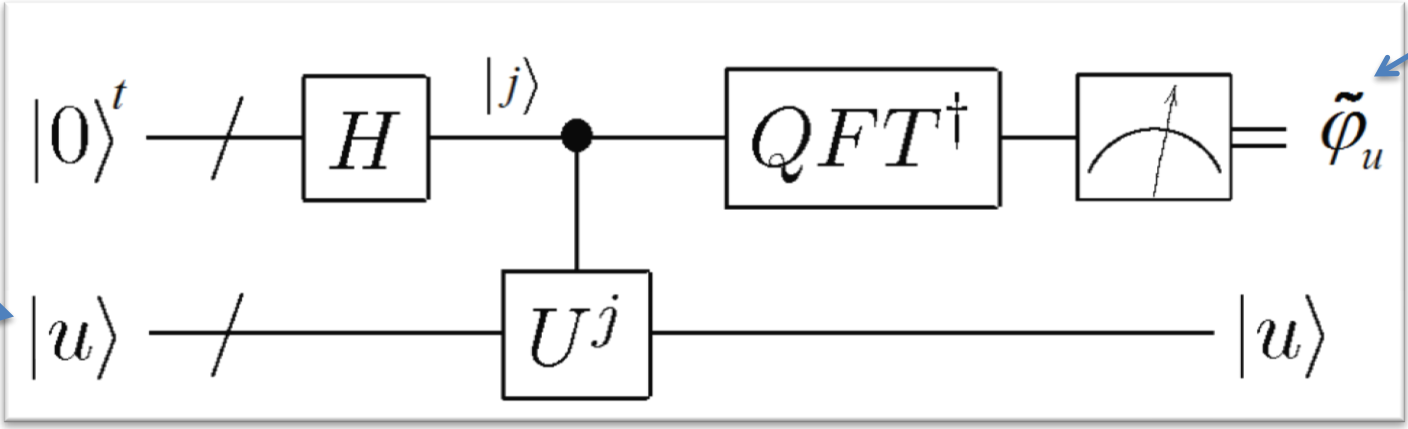
QPE Algorithm– Binary logic case

We first review the binary case:

Let $\varphi_u \approx \frac{\tilde{\varphi}_u}{2^t} = \tilde{\varphi}_1 2^{-1} + \tilde{\varphi}_2 2^{-2} + ..\tilde{\varphi}_{t-1} 2^{-(t-1)} + \tilde{\varphi}_t 2^{-t}$ be the best t bit approximation to φ_u. In short, it is represented as $\varphi_u \approx 0.\tilde{\varphi}_1\tilde{\varphi}_2\tilde{\varphi}_3\ldots\tilde{\varphi}_{t-1}\tilde{\varphi}_t$

Schematic for the QPE Algorithm

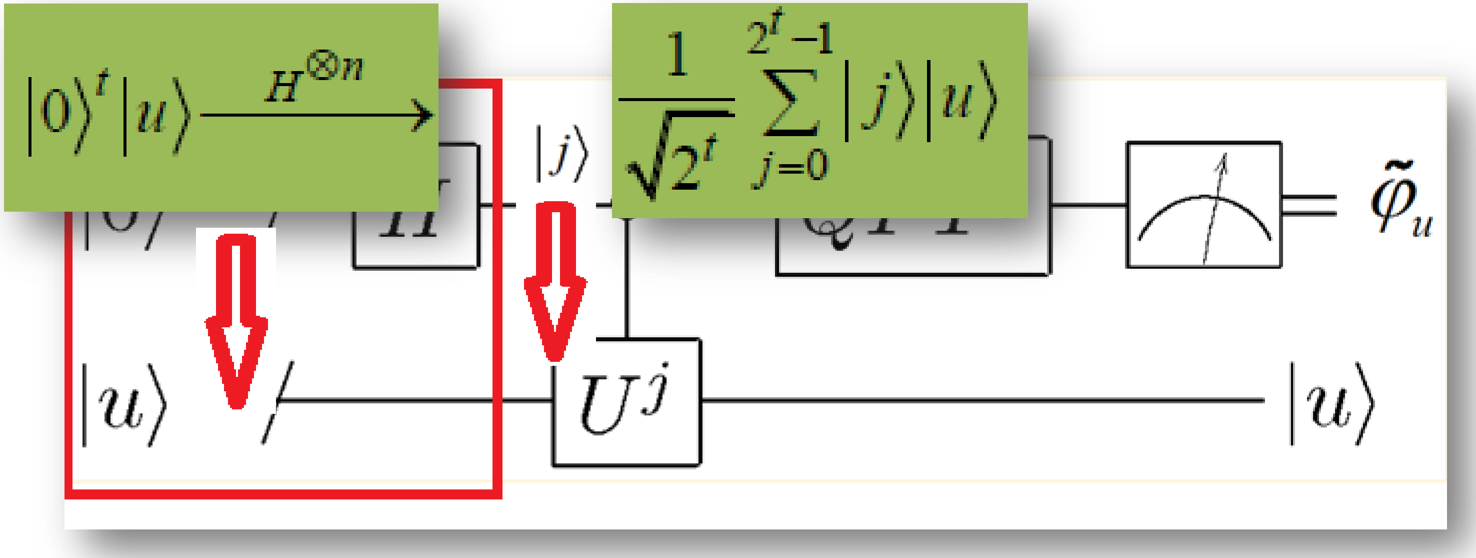
Eigenvector of U



Measure phase in t qubits

QPE : step 1 – Initialization, Binary logic case

$|0\rangle^t$ be the tensor product of t qubits each in the state $|0\rangle$



$$|0\rangle^t|u\rangle \xrightarrow{H^{\otimes n}}$$

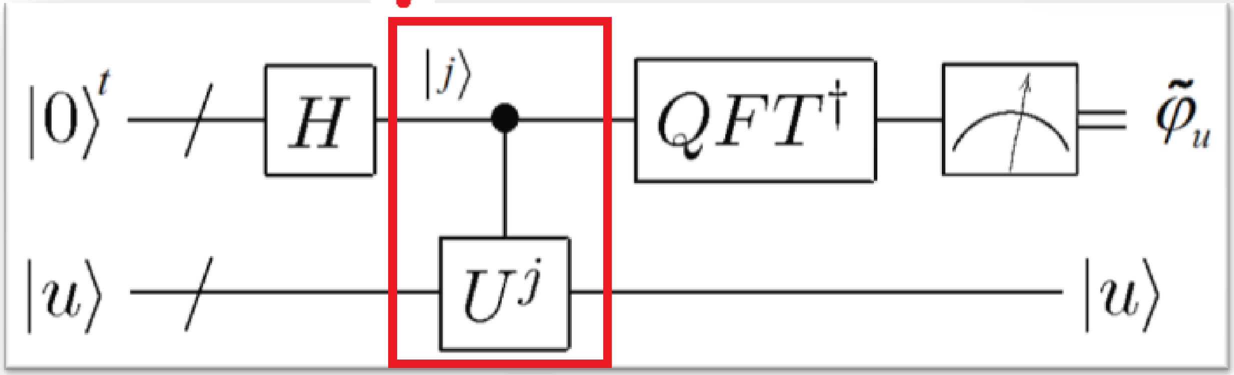
$$\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle|u\rangle$$

QPE : step 2 – Apply the operator *U*

$$\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle|u\rangle \xrightarrow{U^j}$$

$$\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle\left(U^j|u\rangle\right)$$

$$=\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}\left(e^{2\pi i j\frac{\tilde{\varphi}_u}{2^t}}\right)|j\rangle|u\rangle$$



U^j denotes the controlled unitary operator U *controlled on the state vector* $|j\rangle$.

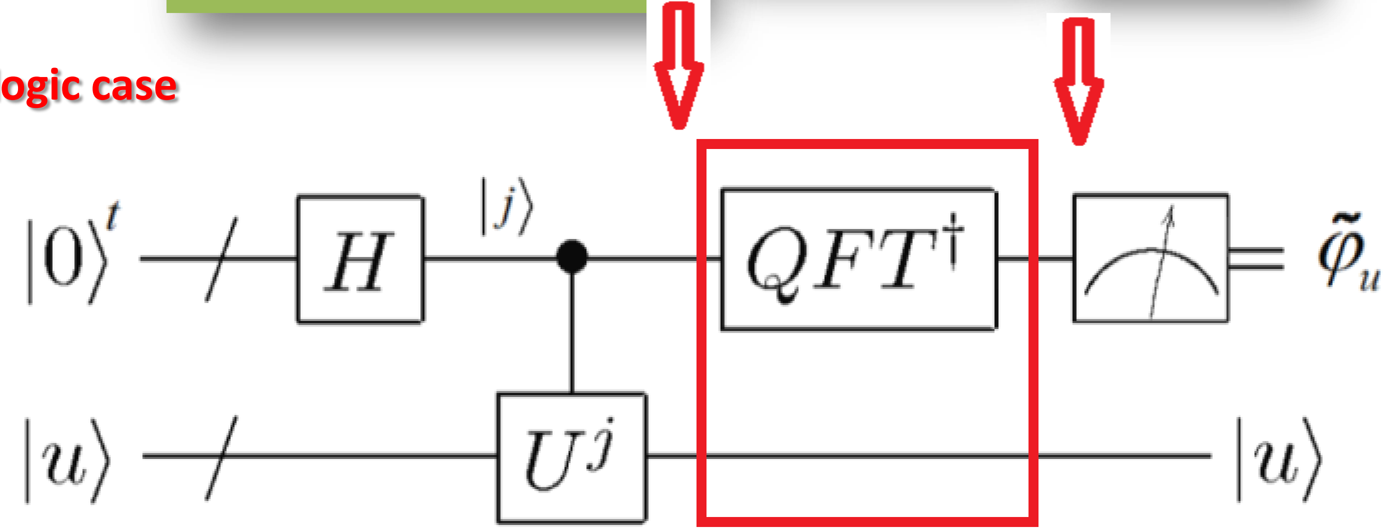
Binary logic case

$$\left(U^j|u\rangle\right)=\left(e^{2\pi i\varphi_u}\right)^j|u\rangle=\left(e^{2\pi i\frac{\tilde{\varphi}_u}{2^t}}\right)^j|u\rangle$$

QPE : step 3 – Apply Inverse QFT

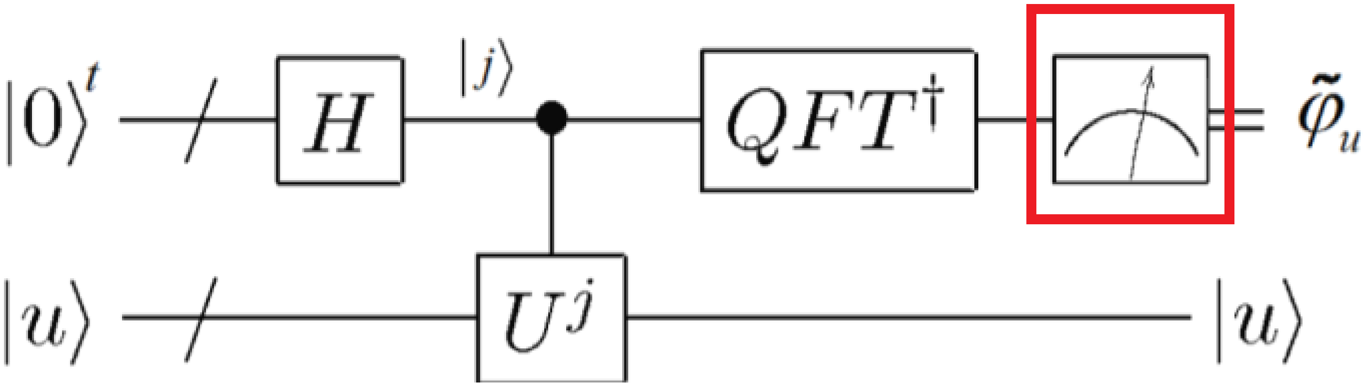
$$\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}\left(e^{2\pi i j\frac{\tilde{\varphi}_u}{2^t}}\right)|j\rangle|u\rangle \xrightarrow{QFT^{\dagger}} |\tilde{\varphi}_u\rangle|u\rangle$$

Binary logic case



QFT Definition: $|j\rangle \rightarrow \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{\frac{2\pi i\, jk}{2^n}}|k\rangle$

QPE : step 4 - Measurement



Binary logic case

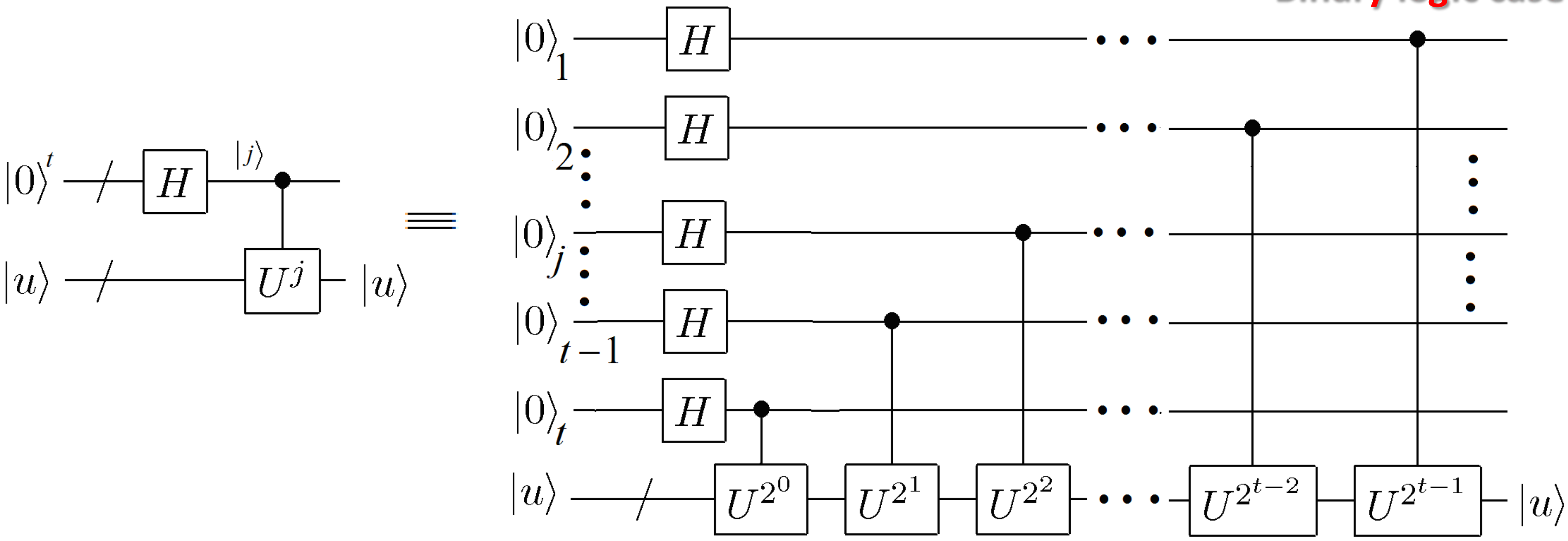
- If the phase ⍰$_u$ is an **exact binary fraction**, we measure the estimate of the phase ⍰$_u$ with probability of 1.
- If not, then we measure the estimate of the phase with a very high probability close to 1.

Quantum circuit for *u*^{j}

If $j = j_1 2^{t-1} + j_2 2^{t-2} + \ldots j_k 2^{t-k} \ldots + j_{t-1} 2^1 + j_t 2^0$.

$$U^j = U^{j_1 2^{t-1}} \times U^{j_2 2^{t-2}} \times \ldots U^{j_k 2^{t-k}} \ldots \times U^{j_{t-1} 2^1} \times U^{j_t 2^0}$$

Binary logic case



The circuit for QFT is well known and hence not discussed.

Multiple-Valued Phase Estimation

QPE – Generalization to MVL

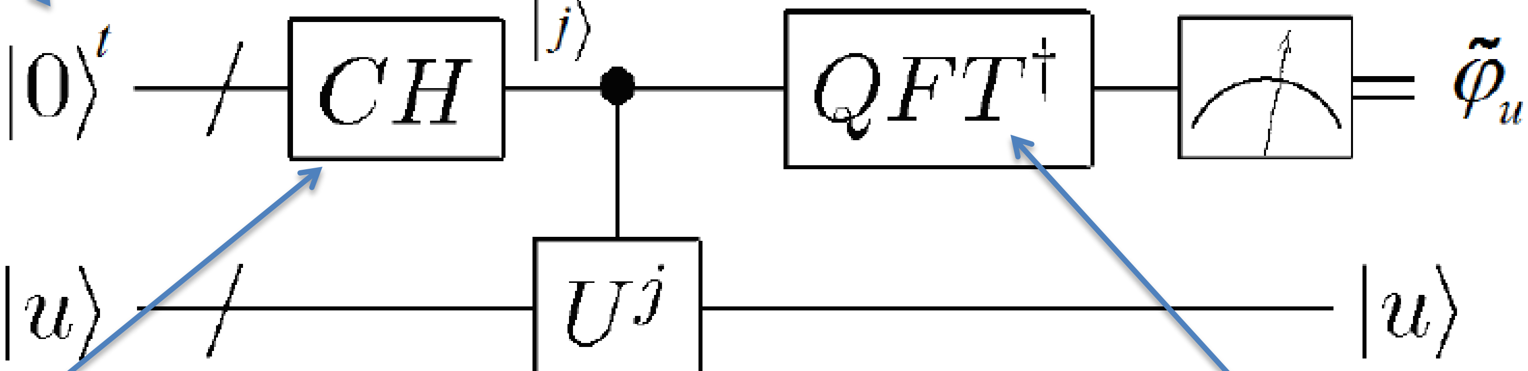
MV logic case

- We represent the phase φ_u as a *d-ary fraction given by* $\varphi_u \approx \frac{\tilde{\varphi}_u}{d^t} = 0.\tilde{\varphi}_1\tilde{\varphi}_2\tilde{\varphi}_3\ldots\tilde{\varphi}_{t-1}\tilde{\varphi}_t$

$$\frac{\tilde{\varphi}_u}{d^t} = \tilde{\varphi}_1 d^{-1} + \tilde{\varphi}_2 d^{-2} + ..\tilde{\varphi}_{t-1} d^{-(t-1)} + \tilde{\varphi}_t d^{-t}$$

We have t qudits for phase

Now we have qudits not qubits



Now we have arbitrary Chrestenson instead Hadamard

Schematic for QPE using Qudits

Now we Inverse QFT on base d, not base 2

Some definitions

QFT and Chrestenson

MV logic case

- The Multivalued logic **QFT** on ***n*** **qudits** is defined as :

$$|j\rangle \xrightarrow{QFT} \frac{1}{\sqrt{d^n}} \sum_{k=0}^{d^n-1} e^{\frac{2\pi i\, jk}{d^n}} |k\rangle$$

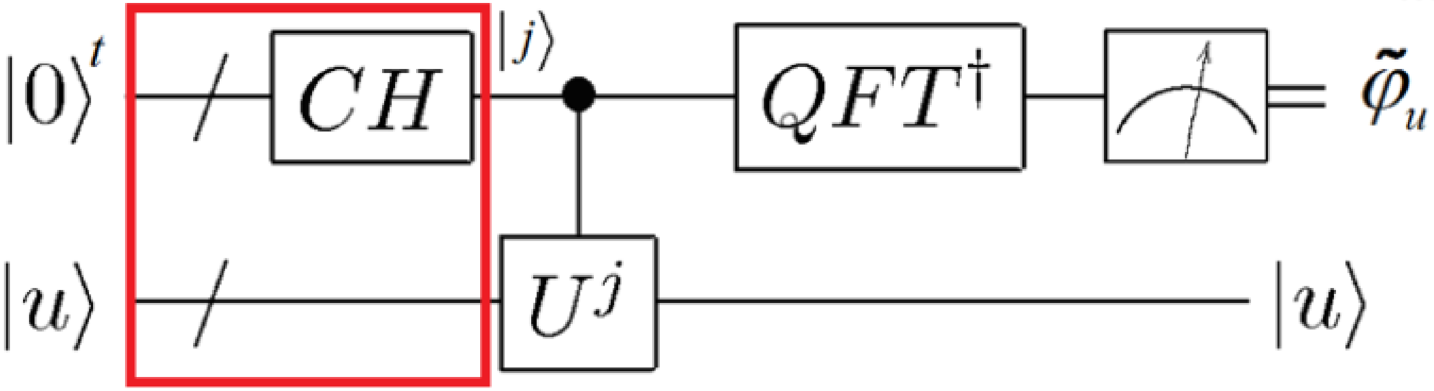
- The action of a **Chrestenson (CH)** gate on a **single qudit** is defined as:

$$CH|x\rangle \rightarrow \frac{1}{\sqrt{d}} \sum_{y\in[0,d-1]} e^{\frac{2\pi ixy}{d}} |y\rangle$$

For *d* = 2, the CH gate reduces to Hadamard gate

MVL QPE : Step 1 - Initialization

MV logic case



$$|0\rangle^{t}|u\rangle \xrightarrow{CH^{\otimes t}} \frac{1}{\sqrt{d^{t}}}\sum_{j=0}^{d^{t}-1}|j\rangle|u\rangle$$

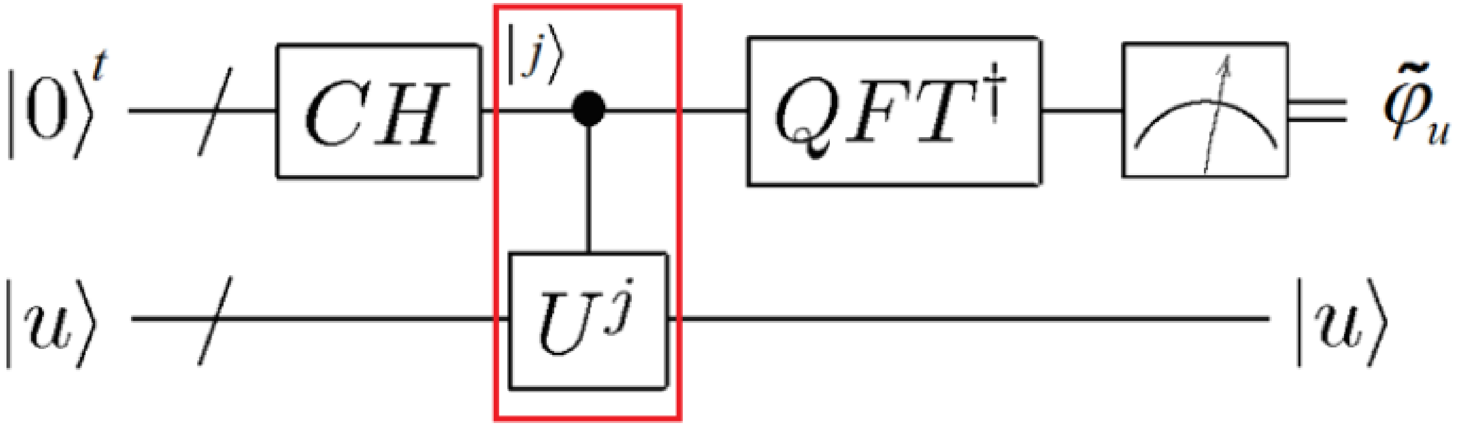
$$\left(CH|0\rangle \rightarrow \frac{1}{\sqrt{d}}\sum_{j\in[0,d-1]}|j\rangle\right)$$

MVL QPE: step 2 – Apply Controlled *u* gate

MV logic case

$$\frac{1}{\sqrt{d^t}}\sum_{j=0}^{d^t-1}|j\rangle|u\rangle \xrightarrow{U^j} \frac{1}{\sqrt{d^t}}\sum_{j=0}^{d^t-1}|j\rangle\left(U^j|u\rangle\right)$$

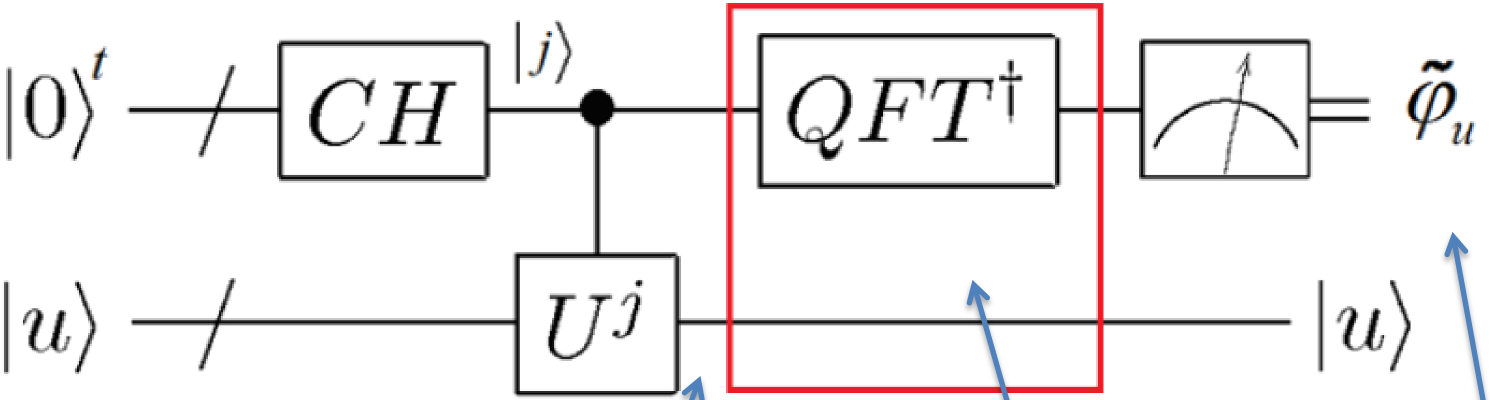
$$=\frac{1}{\sqrt{d^t}}\sum_{j=0}^{d^t-1}|j\rangle\left(e^{2\pi i\varphi_u}\right)^j|u\rangle=\frac{1}{\sqrt{d^t}}\sum_{j=0}^{d^t-1}\left(e^{2\pi ij\frac{\tilde{\varphi}_u}{d^t}}\right)|j\rangle|u\rangle$$



ISMVL 2011, 23-25 May 2011, Tuusula, Finland

MVL QPE: Step 3 – apply iqft

MV logic case



$$\frac{1}{\sqrt{d^t}}\sum_{j=0}^{d^t-1}\left(e^{2\pi i j\frac{\tilde{\varphi}_u}{d^t}}\right)|j\rangle|u\rangle \xrightarrow{QFT^\dagger} |\tilde{\varphi}_u\rangle|u\rangle$$

We apply inverse Quantum Fourier Transform

MVL QPE: Step 4 -- measurement

MV logic case

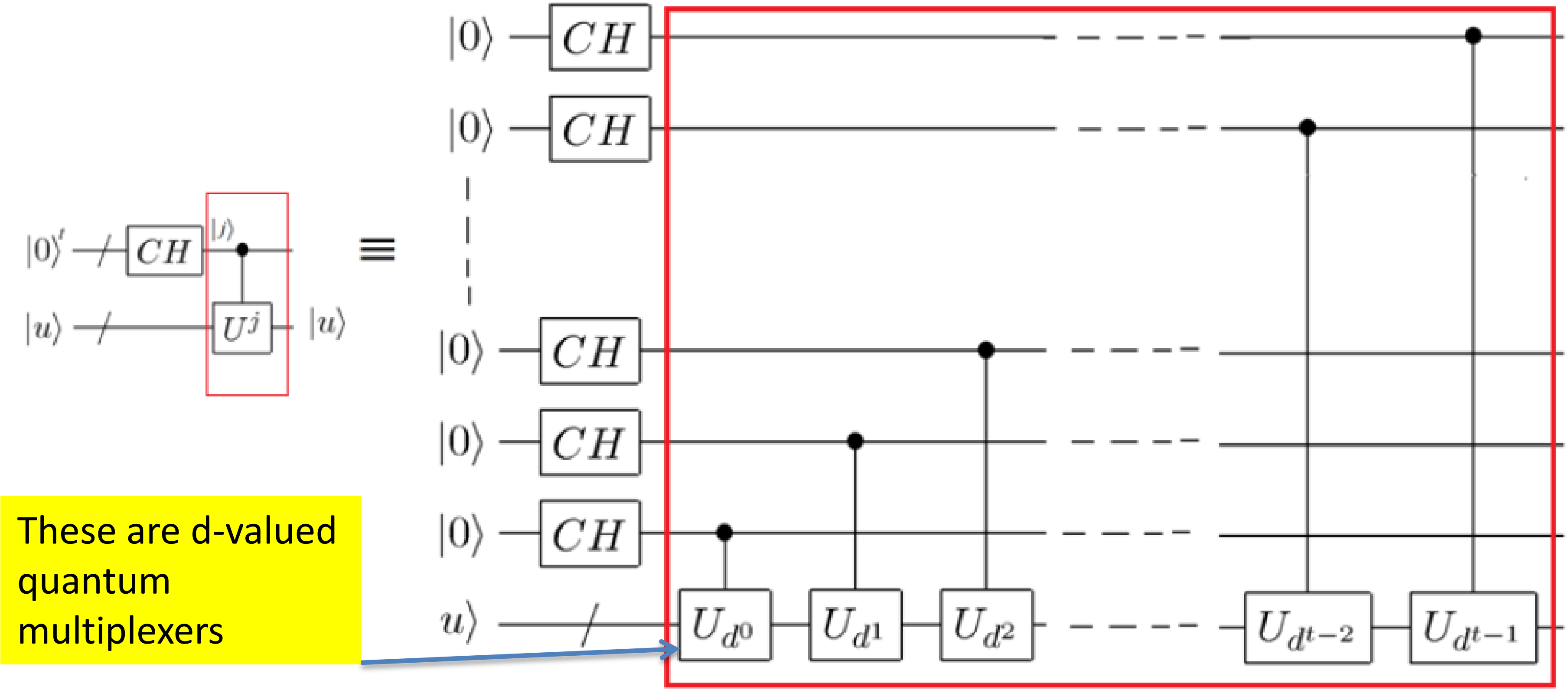
- After making a measurement on the first qudit register, we now get $\tilde{\varphi}_u$ which is an estimate of the phase φ_u

- If φ_u is not an exact *d-ary* fraction then we can only measure the phase with a high success probability close to 1 but not exactly 1.

Quantum circuit for u^j

MV logic case

$$\text{Let } |j\rangle = \bigotimes_{k=1}^{t} |J_k\rangle \quad \Rightarrow U^{j} = U^{\left(\sum_{k=1}^{t} J_k d^{t-k}\right)} = \prod_{k=1}^{t} U^{J_k d^{t-k}}$$

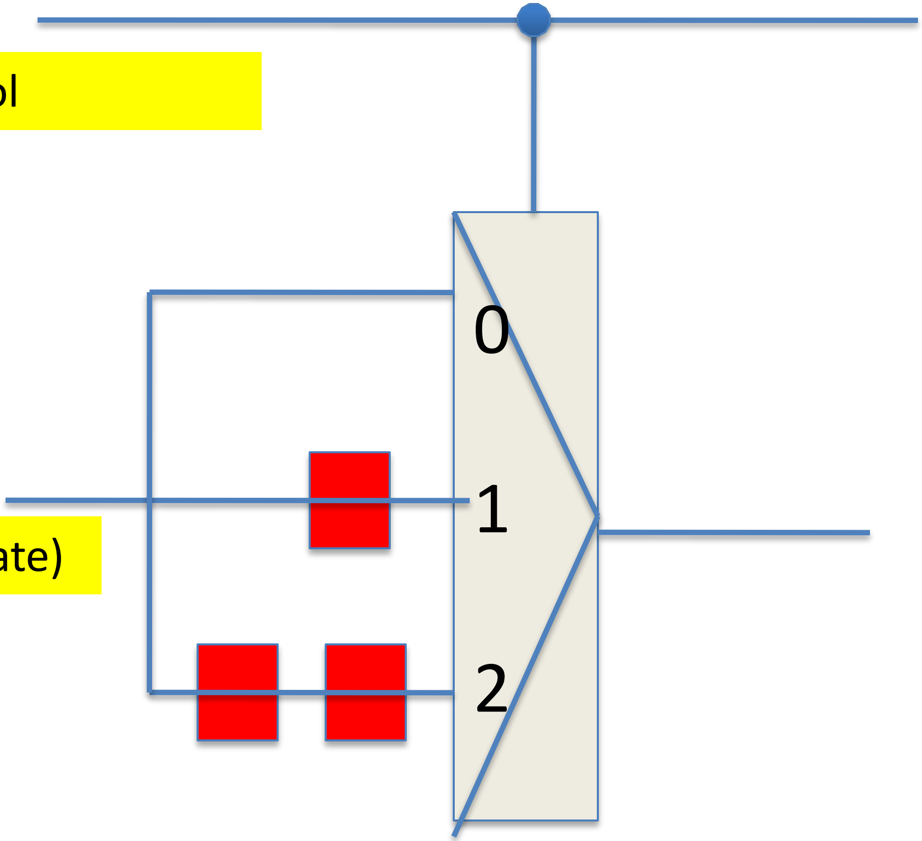


These are d-valued quantum multiplexers

$$U_{d^k} \triangleq diag\left(I, U^{(1d^k)}, U^{(2d^k)}, \ldots, U^{((d-2)d^k)}, U^{((d-1)d^k)}\right)$$

D-valued quantum multiplexers

Case d=3



control

Target (date)

QUANTUM Circuit

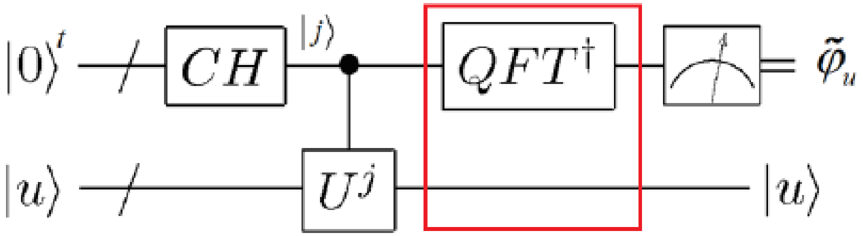
MV logic case

- IQFT can be implemented with a complexity of O(n logn) but the expensive part is implementing higher order powers of U.

- This determines the complexity of the circuit.

Performance of Quantum Phase Estimation

QPE - Performance



MV logic case

$$\frac{1}{\sqrt{d^t}}\sum_{j=0}^{d^t-1}\left(e^{2\pi i j\frac{\tilde{\varphi}_u}{d^t}}\right)|j\rangle|u\rangle \xrightarrow{QFT^\dagger} |\tilde{\varphi}_u\rangle|u\rangle$$

If the phase φ_u is an exact d-ary fraction i.e. $\varphi_u = \frac{\tilde{\varphi}_u}{d^t}$ then after QFT … QPE algorithm gives correct answer with probability of 1.

What if it is not an exact faction ?

$$i.e. \quad \varphi_u = \frac{\tilde{\varphi}_u}{d^t} + \delta \quad \left(0 \le \delta < d^{-t}\right)$$

QPE Performance

binary

- It can be shown that, in the general case, when the phase is not an exact fraction, **QPE succeeds with minimum probability of** $\frac{8}{\pi^2} = 81.5\%$
- **What can we do to increase this success probability very close to 1 ?**
- **Will MVL help in this aspect? YES**

QPE Performance: PHASE ≠ d-ary fraction

- Let’s analyze what happens if the phase is not an exact fraction. Applying the QFT gives a new superposition state

MV logic case

$$\frac{1}{\sqrt{d^t}}\sum_{j=0}^{d^t-1}\left(e^{2\pi i j\frac{\tilde{\varphi}_u}{d^t}}\right)|j\rangle \xrightarrow{QFT^\dagger} \sum_{l=0}^{d^t-1}\alpha_l|l\rangle$$

It is not hard to show that, the probability of measuring an *l given by P(l) is*

$$P(l)=|\alpha_l|^2=\left|\frac{1}{d^t}\sum_{j=0}^{d^t-1}\exp\left(2\pi i\frac{(\varphi_u-j)}{d^t}l\right)\right|^2$$

QPE Performance

- Using the fact that $\varphi_u = \frac{\tilde{\varphi}_u}{d^t} + \delta \quad \left(0 \le \delta < d^{-t}\right)$

We get $P(l) = |\alpha_l|^2 = \left| \frac{1}{d^t} \left(\frac{1 - e^{2\pi i (d^t \delta + (\tilde{\varphi}_u - l))}}{1 - e^{2\pi i (\delta + \frac{\tilde{\varphi}_u - l}{d^t})}} \right) \right|^2$

MV logic case

- Thus after measurement, we get some value l with the probability given above. i.e. this implies $\varphi_u = \frac{l}{d^t}$

- If l is close to $\tilde{\varphi}_u$ then we can say QPE succeeded else QPE fails.
- How close is close ?

QPE : Success Probability lower bound

- It is easy to show that the probability that QPE returns either *l or l+1 such that* $l \le \tilde{\varphi}_u \le l+1$ *is* $\frac{8}{\pi^2} = 81.5\%$ as

$$P(l) = |\alpha_l|^2 = \left| \frac{1}{d^t} \frac{1 - e^{2\pi i \left(d^t \varphi_u - l\right)}}{1 - e^{2\pi i \left(\varphi_u - \frac{l}{d^t}\right)}} \right|^2 \ge \frac{4}{\pi^2}$$

MV logic case

- Although encouraging, the lower bound is not good enough.
- We need SUCCESS PROBABILITY close to 1.
- How to define SUCCESS PROBABILITY ?

Success probability = 1 – failure probability

MV logic case

- Suppose we have a t dit approximation to the phase

$$\varphi_u \approx \frac{\tilde{\varphi}_u}{d^t} = 0.\tilde{\varphi}_1\tilde{\varphi}_2\tilde{\varphi}_3\ldots.\tilde{\varphi}_{t-1}\tilde{\varphi}_t$$

- If we are only interested in a precision of only upto n dits

$$i.e. \quad \varphi_u = \frac{\tilde{\varphi}_u}{d^t} + \delta \quad \left(0 \le \delta < d^{-n}\right)$$

- then as long as QPE returns some l $i.e.$ $\varphi_u = \dfrac{l}{d^t}$ such that the above condition is satisfied, we have a success.

- The error e is

$$e = l - \tilde{\varphi}_u = \delta d^t \quad \text{but} \quad \left(0 \le \delta < d^{-n}\right)$$

$$\Rightarrow e < d^{t-n}$$

Failure probability

- We define the failure probability as

$$\varepsilon = p(|l - \tilde{\varphi}_u| > e) = \sum_{l=0, l \notin [\tilde{\varphi}_u - e, \tilde{\varphi}_u + e]}^{d^t - 1} |\alpha_l|^2 \leq \frac{1}{2(e-1)}$$

- The failure probability has a lower bound and hence the success probability has an upper bound.

$$p(Success) = 1 - \varepsilon > 1 - \frac{1}{2(e-1)}$$

$$where \;\; e = d^{t-n} - 1$$

Success probability: REQUIREMENTS

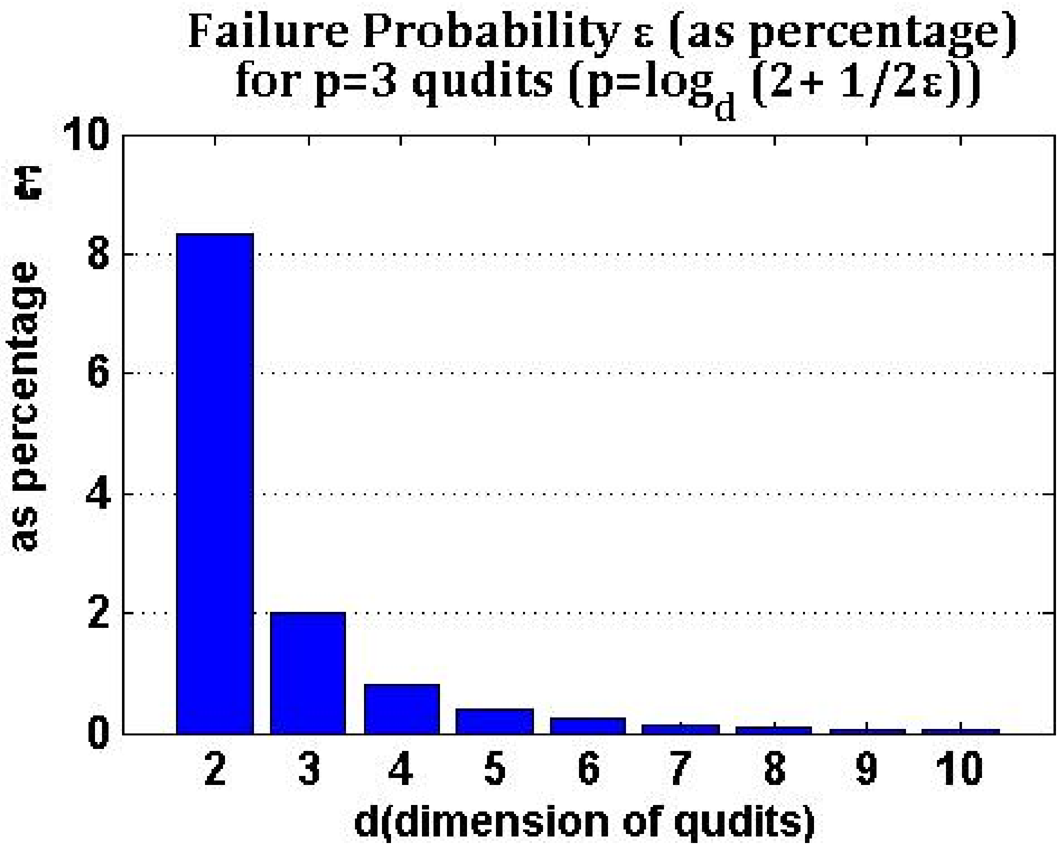
- Thus to achieve phase estimation with a success probability of $1-\varepsilon$ with precision/accuracy up to n dits, we need to use a system with t dits. The value of t is given by

$$t = n + p = n + \log_d \left(2 + \frac{1}{2\varepsilon} \right)$$

- Now we show quantitative results in some graphs

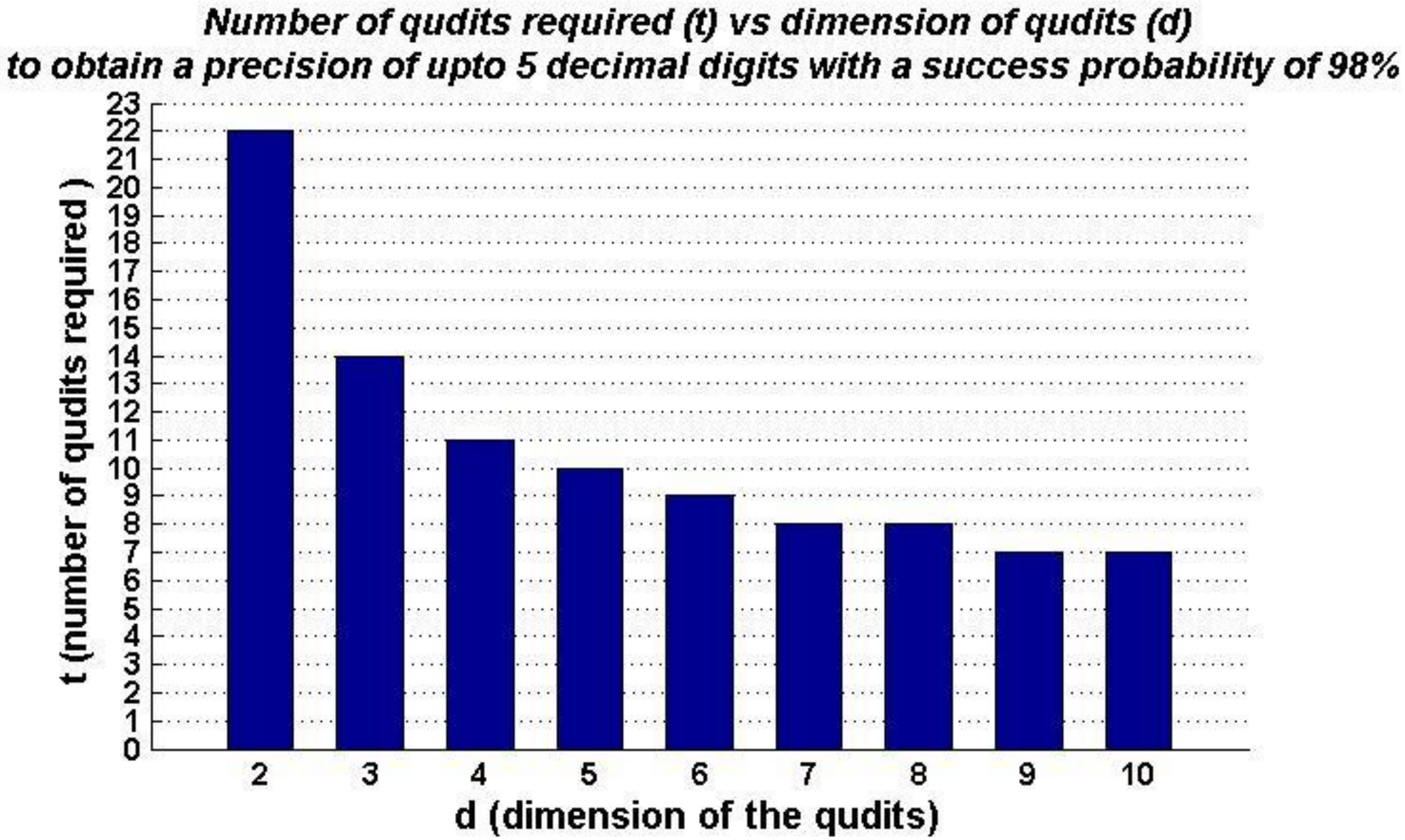
How MVL HELPS

- **Failure probability** decreases exponentially with increase in radix *d* of the logic used

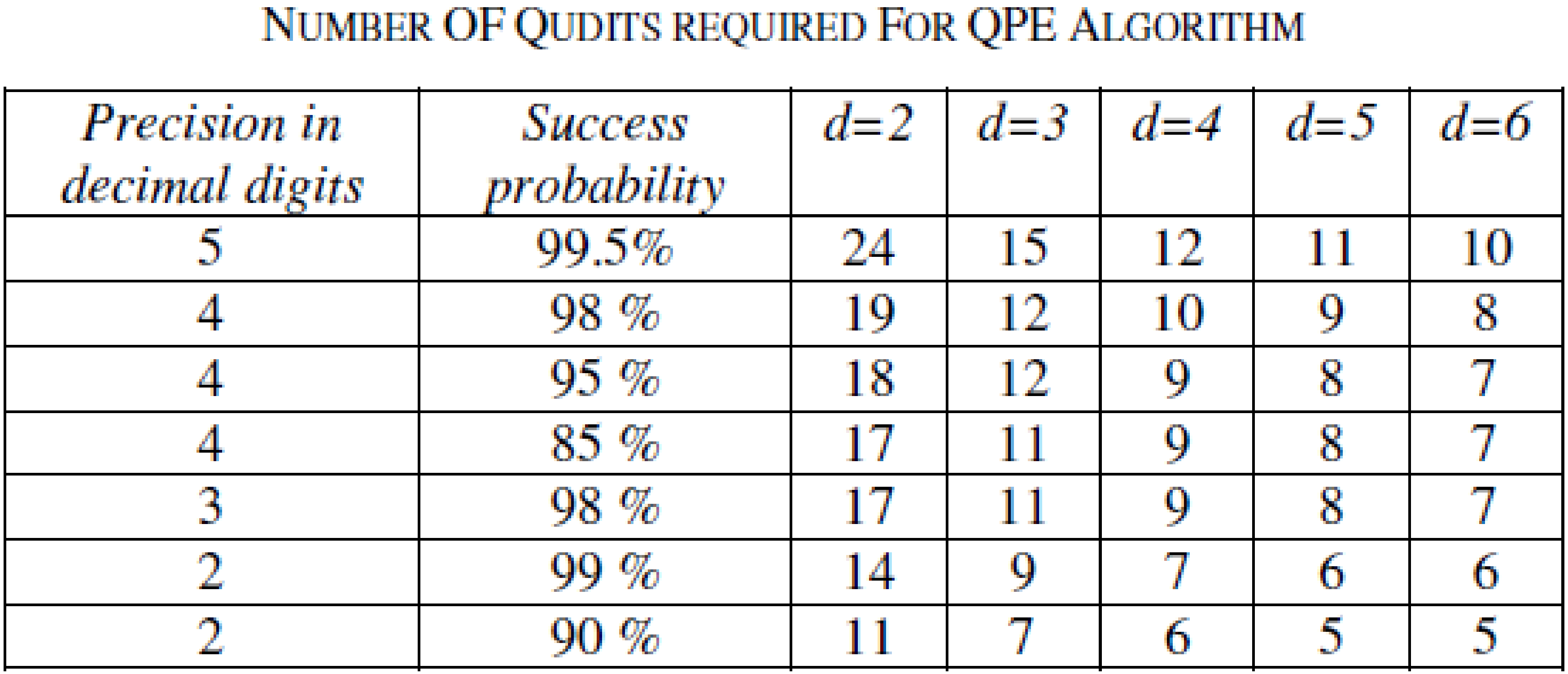


Less number of qudits for a given precision

These are the requirements for a real world problem of calculating molecular energies



More RESULTS



NUMBER OF QUDITS REQUIRED FOR QPE ALGORITHM

Precision in decimal digits	*Success probability*	*d=2*	*d=3*	*d=4*	*d=5*	*d=6*
5	99.5%	24	15	12	11	10
4	98 %	19	12	10	9	8
4	95 %	18	12	9	8	7
4	85 %	17	11	9	8	7
3	98 %	17	11	9	8	7
2	99 %	14	9	7	6	6
2	90 %	11	7	6	5	5

Conclusions

- Quantum Phase Estimation has many applications in Quantum Computing

- MVL is very helpful for Quantum Phase Estimation

- Using MVL causes exponential decrease in the failure probability for a given precision of phase required.

- Using MVL results in signification reduction in the number of qudits required as radix *d* increases

Conclusions 2

- The method creates high power unitary matrices U^k of the original Matrix U for which eigenvector |u> we want to find phase.
- We cannot design these matrices as powers. This would be extremely wasteful
- We have to calculate these matrices and decompose them to gates
- New type of quantum logic synthesis problem: not permutative U, not arbitrary U, there are other problems like that, we found
- This research problem has been not solved in literature even in case of binary unitary matrices U