

Kuantum Fourier Dönüşümünün Yüksek Boyutta Cirq Kullanarak Uygulanması

Implementation of High Dimension Quantum Fourier Transform via Cirq

Osman Semi Ceylan¹ 

Cumali Yaşar² 

¹ Çanakkale Onsekiz Mart Üniversitesi, Bilgisayar Mühendisliği, Türkiye, e-mail: osmansemi.ceylan@comu.edu.tr

² Çanakkale Onsekiz Mart Üniversitesi, Eğitim Fakültesi, Türkiye, e-mail: cumali.yasar@comu.edu.tr

Öz

Kuantum Fourier Dönüşümü (QFT), kuantum hesaplamada çeşitli kuantum algoritmaları için çok önemli bir işlemdir. Bu çalışma, devre tabanlı simülasyon yazılımında yüksek boyutta QFT algoritmasının bir uygulamasını sunmaktadır. İlk olarak QFT algoritması, kuantum sistemlerde matematiksel olarak incelenmektedir. Ardından, kuantum devreler üzerinde yüksek boyutlu QFT algoritması verilmektedir. Son olarak, bu tekniklerin Cirq kuantum devre simülasyon yazılımı üzerinde pratik bir uygulamasını sunuyoruz. Çalışmamız, kuantum devre simülasyonlarında yüksek boyutlu QFT algoritmasının uygulanmasına yönelik değerli bilgiler ve pratik yönergeler sağlar. Böylelikle kuantum yüksek boyuta daha fazla ilgi çekeceği düşünülmektedir.

Anahtar Kelimeler: Kuantum Fourier Dönüşümü, Devre Simülasyonu, Yüksek Boyutta Kuantum Algoritmalar

Abstract

Quantum Fourier Transform (QFT) is a crucial operation for various quantum algorithms in quantum computing. This work presents an implementation of the high-dimensional QFT algorithm in circuit-based simulation software. Firstly, the QFT algorithm is examined mathematically in quantum systems. Then, the high-dimensional QFT algorithm on quantum circuits is given. Finally, we present a practical implementation of these techniques on the Cirq quantum circuit simulation software. Our work provides valuable information and practical guidelines for applying the high-dimensional QFT algorithm in quantum circuit simulations. Thus, it is thought that quantum will attract more attention to the higher dimension.

Keywords: Quantum Fourier Transform, Circuit Simulation, High Dimension Quantum Algorithms

Citation/Atf: CEYLAN, O. S. & YAŞAR, C. (2024). Kuantum Fourier Dönüşümünün Yüksek Boyutta Cirq Kullanarak Uygulanması. *Kuantum Teknolojileri ve Enformatik Araştırmaları*. 2(1):45-49, DOI: 10.70447/ktve.2372

Corresponding Author/ Sorumlu Yazar:
Osman Semi Ceylan
E-mail: osmansemi.ceylan@comu.edu.tr



Bu çalışma, Creative Commons Atif 4.0 Uluslararası Lisansı ile lisanslanmıştır.
This work is licensed under a Creative Commons Attribution 4.0 International License.

1. GİRİŞ

Kuantum Fourier Dönüşümü (QFT)[1], kuantum hesaplama alanında en değerli işlemlerden biridir. Geleneksel formunda QFT, klasik durumların üst üste binmesini temsil eden bir kuantum durumunu zarif bir şekilde frekans alanı karşılığına dönüştürür. Bu dönüşüm, Shor'un tam sayıları asal çarpanlara ayırma algoritması[2] gibi çeşitli kuantum algoritmalarının elde ettiği hızlanmanın temelini oluşturuyor; bunun özellikle kriptografi alanı için önemli etkileri var[3].

QFT algoritmasının geleneksel iki seviyeli sistemlerdeki etkisi derin olsa da daha yüksek boyutlara yayılması büyüleyici ve nispeten keşfedilmemiş bir sınır sunuyor. Yüksek boyutlu QFT algoritması yalnızca kuantum mekaniğinin matematiksel çerçevesini genişletmekle kalmıyor, aynı zamanda kuantum bilgi işlemeyi ilerletmek için zengin fırsatlar da sunuyor.

Yüksek boyutlu QFT algoritmasının önemli bir özelliği, karmaşık kuantum durumlarını ikili veya kübit temsillerinin ötesinde manipüle etme ve analiz etme yetenekleridir. Yüksek boyutlu QFT algoritması, ikiden büyük boyutlara sahip küditler (kuantum rakamları) üzerinde çalışarak, bilgilerin daha kompakt ve anlamlı bir şekilde temsil edilmesini sağlar. Bu, kuantum sistemlerinde gürültünün etkilerini azaltmak için gerekli olan daha verimli kuantum algoritmalarına ve gelişmiş kuantum hata düzeltme kodlarına yol açabilir[4]. Bu nedenle bu çalışmada QFT algoritması Cirq[5] simülasyon yazılımı üzerinde uygulanmasını inceledik.

Bu makale 4 bölümde hazırlanmıştır. İkinci bölümde kuantum yüksek boyut cebiri anlatılmaktadır. Üçüncü bölümde yüksek

boyutta QFT ve devresi verilmektedir. Dördüncü bölümde QFT algoritmasının Cirq yazılımı üzerinde uygulaması bir örnek ile gösterilmiştir. Son bölümde ise bulgularımızdan varılan sonuçlara yer verilmiştir.

2. KUANTUM YÜKSEK BOYUT

Küditler veya kuantum rakamlar, d -boyutlu karmaşık Hilbert uzayındaki (\mathcal{H}) bir kuantum durumunu matematiksel olarak ifade eder. Bu uzaydaki küditlerin ortonormal Fock tabanları eşitlik (1) ile verilmiştir.

$${}_d|f\rangle \in \{|0\rangle, |1\rangle, |2\rangle, \dots, |d-1\rangle\} \quad (1)$$

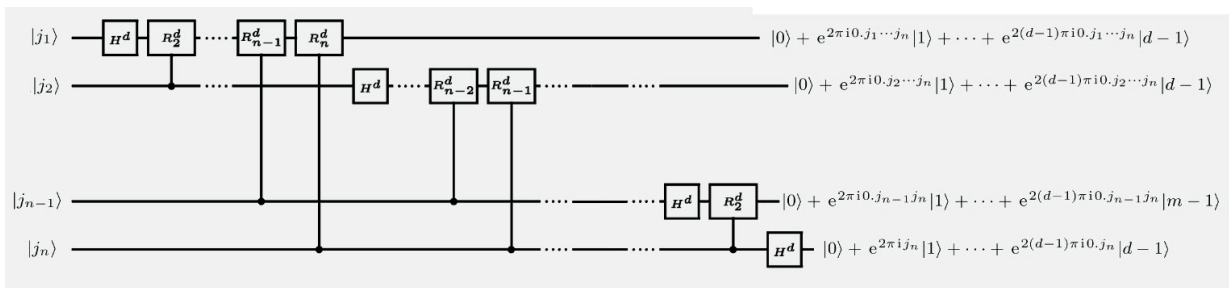
Herhangi bir d -boyuta nicemlenmiş bir kuantum durum ise eşitlik (2)'deki gibi tanımlanmaktadır.

$${}_d|\psi\rangle = \sum_{k=0}^{d-1} \alpha_k |k\rangle \in \mathbb{C}^d \quad (2)$$

Vektörel formda olan bir ${}_d|\psi\rangle$ kuantum durumun özdeğerleri α_k olarak verilir. Geleneksel iki seviyeli sistemlerde olduğu gibi yüksek boyutta da kuantum durum bir Gaussian dağılımı olmak zorundadır.

$$\sum_k \alpha_k \alpha_k^* = 1 \quad (3)$$

Herhangi bir d -boyutta tanımlanan küdit bir kuantum durumu dönüştürmek için kullanılan dönüşüm matrisleri $(d^n \times d^n)$ boyutlarında olmalıdır. Herhangi bir d boyutun Weyl dönüşümleri[6] olarak da bilinen temel dönüşüm matrisleri denklem (4) ile hesaplanmaktadır.



Şekil-1: QFT algoritmasının tasarımsal kuantum devresi[8].

$${}_dU_{pq} = \sum_{k=0}^{d-1} e^{\frac{2\pi i}{d}kp} |(k+q) \bmod d\rangle \langle k|, \quad (4)$$

$$p, q \in \{0, 1, \dots, d-1\}$$

Denklem (4) ile verilen dönüşümler kullanılarak yüksek boyutta kuantum hesaplamada iki seviyeli sistemlerde bulunan dönüşümlerin benzerleri elde edilebilir. Bir örnek olarak kapısının denklem (5) ile iki boyutta ve denklem (6) ile 3 boyutta matrisleri verilmektedir.

$$X = {}_2U_{01} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (5)$$

$${}_3U_{01} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad {}_3U_{02} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad (6)$$

Denklem (6) ile verilen dönüşümlerin 3 boyut kuantum özvektörlere uygulanması denklem (7) ile gerçekleştirilir.

$$\begin{aligned} {}_3U_{01}|0\rangle &= |0 \oplus 1\rangle = |(0+1) \bmod(3)\rangle = |1\rangle \\ {}_3U_{01}|1\rangle &= |0 \oplus 1\rangle = |(1+1) \bmod(3)\rangle = |2\rangle \\ {}_3U_{01}|2\rangle &= |0 \oplus 1\rangle = |(2+1) \bmod(3)\rangle = |0\rangle \end{aligned} \quad (7)$$

Benzer olarak d -boyutta QFT içinde gereken tek küdit kuantum faz dönüşüm kapısının matrisi de eşitlik (8)'de verilmektedir.

$$\text{diag}({}_dRZ_k) = (1, \omega_k, \omega_k^2, \dots, \omega_k^{d-1}), \quad \omega = e^{\frac{2\pi i}{d^k}} \quad (8)$$

3. Yüksek Boyut QFT

Yüksek boyutlarda QFT, iki boyutta olduğu gibi tam süperpozisyon durumunda faz dönüşümleri uygular. Bu nedenle genel Fourier dönüşümü formülü ile herhangi bir d -boyutunda hesaplanabilirler. Bu çalışmada da kullanılan Kuantum Genel Fourier Dönüşümü (QGFT)[7] algoritmasının n küditlik bir $|j\rangle$ durumu üzerindeki etkisinin d -boyutlu formülü aşağıdaki biçimdedir[8]:

$$F(d, N)|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i}{N}jk} |k_d\rangle, \quad N = 2^n \quad (9)$$

Ayrıca denklem (9) kullanılarak -boyutta tam süperpozisyon üreten Hadamard kapısının matrisi $n = 1$ alınarak eşitlik (10)'daki gibi hesaplanabilir[8].

$${}_dH|j\rangle = F(d, d)|j\rangle = \frac{1}{\sqrt{d}} \sum_{k=0}^{d-1} \omega^{jk} |k\rangle \quad (10)$$

Herhangi bir -boyutta QFT uygulanması için denklem (8) ile verilen faz kapısı ve denklem (10) ile verilen Hadamard kapısı kullanılmaktadır.

Bir d -boyutta QFT gerçekleştiren kuantum devre Şekil-1 ile verilmiştir. Devre üzerinde J ile dizilenmiş n adet küditten oluşmaktadır. Şekilden görüleceği üzere QFT algoritması d -boyut ${}_dH$ ve belirli bir desen ile devreye yerleştirilmiş kontrollü ${}_dRZ_k$ kapılarından oluşmaktadır.

İki seviyeli sistemlerin aksine yüksek boyutlu hesaplamada çok değerli kontrol kapıları (MVCG) bulunmaktadır. Bu kapılar yalnız değil, istenilen diğer özvektörlere koşullu kapıların uygulanmasını sağlamaktadır. Herhangi bir seçilen d -boyut hesaplamada koşullu bir kapı için koşul değerleri kümesi ${}_dG \subset \{0, 1, \dots, d-1\}$ gibi boş olmayan bir altküme olmalıdır. QFT algoritması yüksek boyutta gerçekleştirilmesine bu çeşit kontrollü kapılar kullanılmaktadır.

4. QFT Algoritmasının Cirq Uygulaması

Cirq üzerinde yüksek boyutta kapı tabanlı kuantum hesaplama simülasyonları elverişli bir platformdur. Ayrıca yüksek boyutta kodlama yapabilmek için gerekli bazı araçları da hazır sunmaktadır.

4.1. Yüksek Boyutta Matrislerin Hesaplanması

QFT algoritması için Hadamard ve koşullu faz kapıların Cirq üzerinde tanımlanması gerekmektedir. Bir kuantum kapının tanımlanması için d -boyutlu kapının matrisi hesaplanmalıdır. Hadamard için denklem (10) ile verilen eşitlik kullanılabilir. Aşağıda $d = 3$ için Hadamard matrisi gösterilmektedir.

$${}_3H = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\ 1 & e^{\frac{2\pi i}{3}} & e^{\frac{4\pi i}{3}} \\ 1 & e^{\frac{4\pi i}{3}} & e^{\frac{8\pi i}{3}} \end{pmatrix} \quad (11)$$

Benzer şekilde $d = 3$ için ${}_3RZ_1$, ${}_3RZ_2$, ${}_3RZ_3$ için faz kapıları ise aşağıdaki biçimde matrisler olmaktadır.

$${}_3RZ_1 = {}_3Z = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{\frac{2\pi i}{3}} & 0 \\ 0 & 0 & e^{\frac{4\pi i}{3}} \end{pmatrix} \quad (12)$$

$${}_3RZ_2 = {}_3S = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{\frac{2\pi i}{9}} & 0 \\ 0 & 0 & e^{\frac{4\pi i}{9}} \end{pmatrix} \quad (13)$$

$${}_3RZ_3 = {}_3T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{\frac{2\pi i}{27}} & 0 \\ 0 & 0 & e^{\frac{4\pi i}{27}} \end{pmatrix} \quad (14)$$

4.2. Yüksek Boyutta Kapıların Tanımlanması

QFT algoritmasında kullanılan bu matrislerin kapı olarak Cirq yazılımı üzerinde tanıtılması gerekmektedir. Cirq üzerinde matris hazırlamak için Şekil-2'de verilen kod ile yazılımsal sınıf oluşturulmalıdır.

```
class Gate(cirq.Gate):
    def __init__(self, unitary_matrix: dimension: int, diag_info):
        self.unitary_matrix = unitary_matrix
        self.dimension = dimension
        self.qubit_count = int(qcount)
        self.gate_name = diag_info
        super().__init__()

    def qid_shape(self) -> typing.Tuple[int]:
        return (self.dimension,) * self.qubit_count

    def num_qubits(self) -> int:
        return self.qubit_count

    def unitary(self) -> numpy.ndarray:
        return self.unitary_matrix

    def circuit_diagram_info(self, args) -> cirq.CircuitDiagramInfo:
        return self.gate_name

    def __str__(self) -> str:
        return f'{self.unitary_matrix}'

@property
def transform_matrix(self) -> numpy.ndarray:
    return self.unitary_matrix
```

Şekil-2: Cirq üzerinde tanımlı kapı nesnesi üreten sınıf.

Şekil-2 ile verilen yazılımsal sınıf, matrisi boyut miktarı ve kapının ismini parametre olarak almakta ve Cirq üzerinde devrelerde kullanılması için hazırlanmıştır.

4.3. QFT Devresi Oluşturma

QFT algoritmasının Cirq üzerinde uygulanması için öncelikle kuantum devre hazırlanmalıdır. Sonrasında QFT için gerekli kapılar aşağıdaki biçimde verilen sınıf kullanılarak kodlanmalıdır.

```
circuit = cirq.Circuit()
qudits = cirq.LineQid.range(3, dimension=3)

hadamard_gate = Gate(general_Hadamard(dim=3), dimension=3,
diag_info='H')
swap_gate = Gate(swap(dim=3), dimension=3, diag_info=('x', 'x'))
z_gate = cirq.ZPowGate(exponent=1/numpy.power(2, 0),
dimension=3)
s_gate = cirq.ZPowGate(exponent=1/numpy.power(2, 1),
dimension=3)
t_gate = cirq.ZPowGate(exponent=1/numpy.power(2, 2),
dimension=3)
```

Şekil-3: Cirq üzerinde devre ile QFT için gereken kapıların tanımlanmasını içeren kod.

Şekil-3 ile örnek olarak boyut sayısı $d = 3$, küdit sayısı alınarak oluşturulan kuantum devresi ve QFT uygulanması için gereken Hadamard ve faz kapıları tanımlamalarını içerir.

```
z_gate = z_gate.controlled(num_controls=1, control_values=((1, 2),),
control_qid_shape=(3,))
```

Şekil-4: Cirq üzerinde koşullu kapı oluşturan kod.

Bu faz kapıların QFT algoritmasında kullanılması için koşullu yapılması gerekmektedir. Çok değerli koşullu kapılar Şekil-4 ile verilen kod ile eklenmektedir.

Elde edilen kapılar Cirq üzerinde devreye eklenerek QFT algoritması tamamlanır. Şekil-5a ile hazırlanan QFT algoritmasının boyut sayısı $d = 3$ ve küdit sayısı $n = 3$ olduğu durumda Cirq üzerinde hazırlanmış devresi verilmektedir. Şekilden görüleceği üzere ilk adımda SWAP kapısı sonra tüm küditlere Hadamard kapısı ve sıralı biçimde S ve T devreye koşullu olarak eklenmiştir. En son olarak yüksek boyutta ölçüm kapısı eklenmiştir. QFT devresinde çok değerli koşullu kapıların koşul değerleri hepsi için (1,2) olarak belirlenmiştir.

4.4. QFT Devresi Simülasyonu

Bu bölümde oluşturulan d -boyut QFT algoritması devresi örnekte verildiği biçimde simülasyon sonuçları verilmektedir. Denklem (10) ile verilen eşitlik hesaplandığında ölçüm sonuçlarında eşit olasılıkta $d^n = 3^3 = 27$ adet durum vektörü beklenmektedir.

Yapılan simülasyon sonucunda Şekil-5c ile görüleceği üzere beklenen sonuçlar elde edilmiştir. Yapılan simülasyonda ölçülen özvektörler $|000\rangle, |001\rangle, |002\rangle, |100\rangle, \dots, |222\rangle$ olmak üzere toplam 27 adet ve eşit olasılıkta oldukları bulunmuştur.

Bu çalışmada yapılan bir diğer simülasyonda ise QFT ve ters QFT algoritması peşi sıra uygulanarak elde edilen sonuçlar incelenmiştir. Şekil-5b'den görülebileceği üzere kuantum durum başlangıç durumuna olan $|0\rangle$ durumuna geri dönmüştür.

5. SONUÇ

Yüksek boyutlu kuantum hesaplama, daha az küdit ile daha fazla bilgi depolayabilmesi, devre karmaşıklığını azaltması, algoritmaların verimliliğini artırabilmesi ve gürültüye karşı hassasiyeti azaltabilmesi gibi çeşitli avantajları sayesinde günümüzde giderek daha fazla ilgi görüyor. Bu nedenle bu çalışmada yapılan QFT algoritmasının yüksek boyutta Cirq yazılımında kodlanabileceği alanın literatürüne yaptığı katkı bakımından yüksek boyuta ilgiyi arttıracığı düşünülmektedir.

KAYNAKÇA

Nielsen, Michael A., Isaac L. Chuang. *Quantum computation and quantum information*, Cambridge university press, 2010. pp.

Shor, Peter W. *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM review 41.2 (1999): 303-332.

Stetsyuk, P. I. *Theory and software implementations of Shor's algorithms*, Cybernetics and Systems Analysis 53.5 (2017): 692-703.

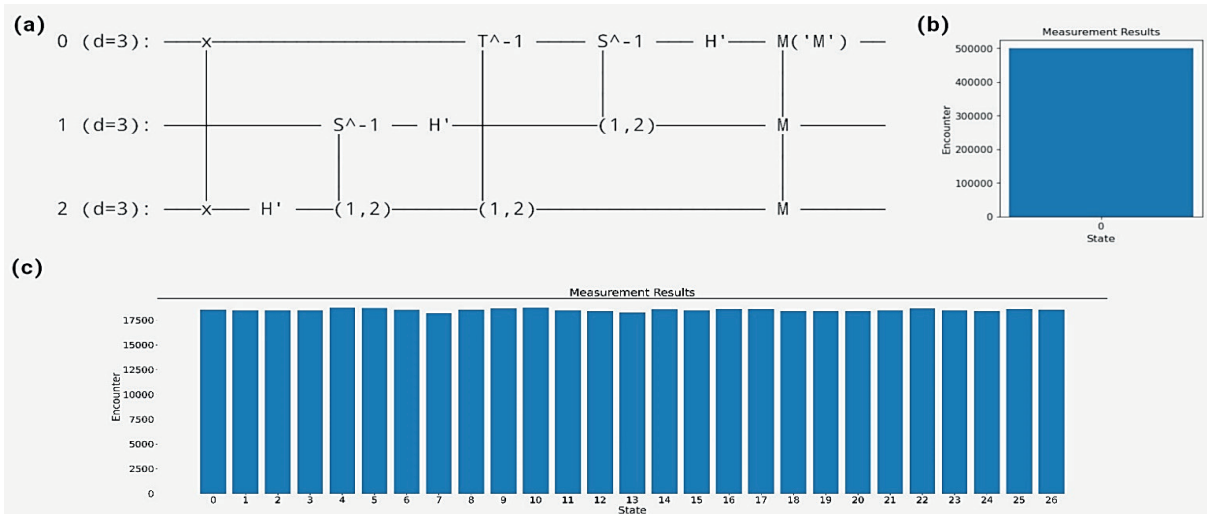
Cozzolino, D., Da Lio, B., Bacco, D., & Oxenløwe, L. K. *High-dimensional quantum communication: benefits, progress, and future challenges*, Advanced Quantum Technologies, 2(12), 1900038.

Cirq Developers. *Cirq*. See full list of authors on Github: <https://github.com/quantumlib/Cirq/graphs/contributors>

Patera, J., and H. Zassenhaus. *The Pauli matrices in n dimensions and finest gradings of simple Lie algebras of type A_{n-1}* , Journal of Mathematical Physics 29.3 (1988): 665-673.

Kitaev, Yu A. *Quantum measurements and the Abelian Stabilizer Problem*, Quant-ph/9511026 (1995).

Cao, Ye, et al. *Quantum fourier transform and phase estimation in qudit system*, Communications in Theoretical Physics 55.5 (2011): 790.



Şekil-5: (a) QFT algoritmasının Cirq üzerinde devre modelinin çıktısı. (b) QFT ve ardından gelen ters QFT algoritmasını içeren kuantum devrenin simülasyon sonucu. (c) QFT algoritmasının kuantum devre simülasyonu sonucu.