

**KARADENİZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**YÜKSEK PERFORMANSLI KUANTUM HESAPLAMA SİMÜLASYONLARI**

**YÜKSEK LİSANS TEZİ**

**Fathelrhman Mohammed ABKER MOHAMMED**

**AĞUSTOS 2019  
TRABZON**



**KARADENİZ TEKNİK ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**YÜKSEK PERFORMANSLI KUANTUM HESAPLAMA SİMÜLASYONLARI**

**Fatmelrhan Mohammed ABKER MOHAMMED**

**Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde**  
**"BİLGİSAYAR YÜKSEK MÜHENDİSİ"**  
**Unvanı Verilmesi İçin Kabul Edilen Tezdir.**

**Tezin Enstitüye Verildiği Tarih : 22 / 07 / 2019**

**Tezin Savunma Tarihi : 08 / 08 / 2019**

**Tez Danışmanı : Dr. Öğr. Üyesi İbrahim SAVRAN**

**Trabzon 2019**

KARADENİZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalında  
Fathelrhman Moham med ABKER MOHAMMED Tarafından Hazırlanan

YÜKSEK PERFORMANSLI KUANTUM HESAPLAMA SİMÜLASYONLARI

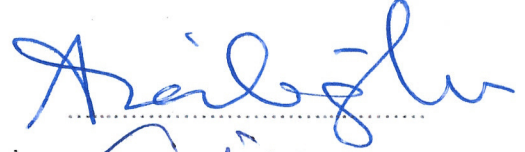
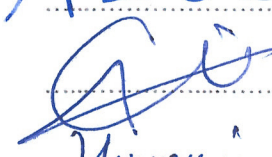

başlıklı bu çalışma, Enstitü Yönetim Kurulunun 23 / 07 / 2019 gün ve 1813 sayılı  
kararıyla oluşturulan jüri tarafından yapılan sınavda  
YÜKSEK LİSANS TEZİ  
olarak kabul edilmiştir.

Jüri Üyeleri

Başkan : Prof. Dr. Abdulsamet HAŞILOĞLU

Üye : Dr. Ogr. Uye. Ibrahim SAVRAN

Üye : Dr. Ogr. Uye. Hüseyin PEHLİVAN

  
.....  
  
.....  
  
.....

Prof. Dr. Asim KADIOĞLU

Enstitü Müdürü

## ÖNSÖZ

Bu Tez Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Bölümü Ana Bilim Dalı'nda Yüksek Lisans Tezi olarak hazırlanmıştır.

Türkiye'de eğitim görmemi sağlayan ve tezin gerçekleşmesi için benden desteğini bir an esirgemeyen Yurtdışı Türkler ve Akraba Topluluklar Başkanlığı (YTB) teşekkürü bir borç bilirim.

Bu çalışmanın gerçekleştirilmesinde, iki yıl boyunca değerli bilgilerini bizlerle paylaşan, kullandığı her kelimenin hayatıma kattığı önemini asla unutmayacağım saygıdeğer danışman hocam; Dr.Öğr.Üyesi İbrahim SAVRAN'a, çalışmam boyunca benden bir an olsun yardımlarını esirgemeyen arkadaşlarım ve çalışma esnasında tüm zorlukları benimle taşıyan ve hayatımın her evresinde bana destek olan değerli Annem, sevgili babam, abilerim ve ablalarıma sonsuz teşekkürlerimi sunarım.

Tez çalışmamda, planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sayın hocalarıma sonsuz teşekkür ederim.

Fathelrhman Mohammed ABKER MOHAMMED  
Trabzon 2019

## TEZ ETİK BEYANNAMASI

Yüksek Lisans Tezi olarak sunduğum "Yüksek Performanslı Kuantum Hesaplama Simülasyonları" başlıklı bu çalışmayı baştan sona kadar danışmanım Dr. Öğr. Üyesi İbrahim SAVRAN'ın sorumluluğunda tamamladığımı ve verileri kendim topladığımı, deneysel çalışmaları IBM araştırma laboratuvarında internet üzerinden yaptığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma süresinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 08/08/2019



Fathelrhman Mohammed ABKER MOHAMMED

1 Haziran 2019

# İÇİNDEKİLER

## Sayfa No

ÖNSÖZ.....	III
TEZ ETİK BEYANNAMASI.....	IV
İÇİNDEKİLER.....	V
ÖZET.....	VII
SUMMARY.....	VIII
ŞEKİLLER DİZİNİ.....	IX
TABLolar DİZİNİ.....	XI
SEMBOLLER DİZİNİ.....	XII
1. GENEL BİLGİLER.....	1
1.1. Giriş.....	1
1.2. Çalışmanı Amacı ve Önemi.....	3
1.3. Yüksek Performans Hesaplama.....	3
1.4. Kuantum Mekaniğin Temelleri.....	3
1.4.1. Kuantum Fizik.....	3
1.4.2. Fotonlar ve Işık.....	4
1.4.3. Işı.....	4
1.4.4. Süperpozisyon.....	5
1.4.5. Dolanıklık (Entanglement).....	5
2. KUANTUM HESAPLAMAYA GİRİŞ.....	6
2.1. Kuantum Bit.....	6
2.2. Çok Kubitli Sistemler: Kuantum Registerlar.....	8
3. KUANTUM KAPILAR.....	10
3.1. Tek Kubit Kapılar.....	10
3.1.1. Pauli Kapıları.....	10
3.1.1.1. Not Kapı ( X Kapı ).....	10
3.1.1.2. Pauli Y Kapısı.....	11
3.1.1.3. Z Kapı.....	12
3.1.2. Hadamard Kapısı (H).....	13

3.2.	Çok Kubitli Kapıları.....	14
3.2.1.	Kontrollü NOT Kapı .....	15
3.2.2.	Swap Kapı .....	17
3.2.3.	Toffoli Kapı (CCNOT) .....	18
3.2.4.	Fredkin Kapı (CSWAP).....	19
4.	KUANTUM PROGRAMLAMA DİLLERİ .....	22
4.1.	Gösterimler.....	22
4.2.	Sözdizimi .....	22
4.3.	QUIRK Kuantum Simulasyon Sistemi .....	23
4.4.	Kuantum Bilgi Bilim Kıt(QISKIT).....	23
4.4.1.	Pauli-X Kapısının ( kuantum NOT) Kodu.....	25
4.4.2.	Grover Algoritmasının Adımları .....	26
4.4.2.1.	Brute Force Arama .....	27
4.4.2.2.	Grover Arama Adımları .....	28
5.	SONUÇLAR .....	37
6.	ÖNERİLER.....	38
7.	KAYNAKLAR .....	39
	ÖZGEÇMİŞ	

## Yüksek Lisans Tezi

### ÖZET

#### YÜKSEK PERFORMANSLI KUANTUM HESAPLAMA SİMÜLASYONLARI

Fathelrhman Mohammed ABKER MOHAMMED

Karadeniz Teknik Üniversitesi  
Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı  
Danışman: Dr. Öğr. Üye. İbrahim SAVRAN  
2019, 41 Sayfa

Kuantum hesaplama, Verileri işlemek, düzenlemek, depolamak ve benzeri işlemleri yapmak için kuantum fiziğini kullanan kuantum bilgisayarlarının kullanımı. Klasik bilgisayar, bit denilen verilerle ilgilenir. Öte yandan, kuantum bilgisayarları kuantum bitleri üzerinde çalışmaktadır. kubit,  $|0\rangle$  ve  $|1\rangle$  olarak ifade edilen temel vektörlerle temsil edilir. Kuantum hesaplamada, bir kuantum mantık kapı (kuantum kapı), kubitler üzerinde çalışan temel bir kuantum devresidir. Onlar kuantum devrelerin yapı taşlarıdır. Bazı kapılar tek bir kubit çalışır. Ayrıca, Toffoli ve Kontrollü NOT kapıları gibi birden fazla kubitler üzerinde hareket eder. Pauli kapıları, Hadamard, tek yuvalı kapıların örnekleridir.

Bilindiği gibi dünyada toplu kullanıma açık bir şekilde sunulan kuantum bilgisayarlar çok sınırlıdır. Bunların arasında IBM tarafından kullanıma sunulan iki adet 5 kubitlik sistemler bulunmaktadır. Aynı şekilde IBM tarafından geliştirilen QISKIT sistemi ile bu fiziksel sistemlerde test kodları denenebilmektedir. Bilindiği gibi 5 kubit bir sistemde sağlıklı bir kuantum sistem sentezlenemez. Bu nedenle QU Touch şirketinin bir ürünü olan QX gibi bazı simülatör kullanarak kuantum kapılarını simüle edebiliriz. Böylece klasik bilgisayarlarda benzetmeye çalışmaktadır.

**Anahtar Kelimeler:** Kuantum Hesaplama; Dolanıklık; Kuantum Kapıları; Kubit; Süperpozisyon; Yüksek Performanslı Hesaplama .



## Master Thesis

### SUMMARY

#### HIGH PERFORMANCE QUANTUM COMPUTING SIMULATIONS

Fathelrhman Mohammed ABKER MOHAMMED

Karadeniz Technical University  
The Graduate School of Natural and Applied Sciences  
Computer Engineering Graduate Program  
Supervisor: Asst. Prof. Dr. Ibrahim SAVRAN  
2019, 41 Pages

Quantum computing, The use of quantum computers that use quantum physics to process data, arrange, store and so on. Classical computer deals with data called bits. On the other hand, quantum computers operate on quantum bits (qubits). Qubits are represented by basis vectors expressed as  $|0\rangle$  and  $|1\rangle$ . In quantum computing, a quantum logic gate (quantum gate) is a basic quantum circuit operating on qubits. They are the building blocks of quantum circuits. While some gates operate on a single qubit. Also, some quantum gates such as Toffoli and Controlled NOT gate manipulate multiple qubits. Pauli gates, Hadamard are examples of single-qubit gates.

Because the lack of resources, we cannot handle our huge applications on physical quantum computers. So we are able to simulate on classic computers. There are some simulator tools such as QISKIT that developed by IBM and QX by QU Touch company.

In this research, we used QISKIT simulator which is an open-source platform. There are Education and Business options for QISKIT. We can install It on a local machine or we access online and it includes a code editor (QASM) and a graphical user interface.

**Keywords:** Quantum Computing; Entanglement; Quantum Gates; Superposition; Qubits; High Performance Computing.

## ŞEKİLLER DİZİNİ

### Sayfa No

Şekil 1. Dik açılarda (URL-8, 2019). .....	4
Şekil 2. Klasik bit .....	6
Şekil 3. Kubit, bir atomdaki iki elektronik seviye tarafından temsil edilir (Nielsen ve Chuang, 2002). .....	7
Şekil 4. Bir kubitin Bloch küre üzerinde gösterimi (Nielsen ve Chuang, 2002). .....	8
Şekil 5. X kapısının Grafik sembolü (Hayes, 2003). .....	11
Şekil 6. Y kapı Grafik sembolü .....	12
Şekil 7. Z kapısının sembolü .....	13
Şekil 8. Hadamard kapısının sembolü .....	14
Şekil 9. Çok kubitli kapı örneği (Nielsen ve Chuang, 2002). .....	15
Şekil 10. CNOT kapı Grafik sembolü .....	16
Şekil 11. SWAP kapısının Grafik sembolü (Williams, 2010). .....	17
Şekil 12. CCNOT kapısının sembolü .....	19
Şekil 13. CSWAP kapısının sembolü .....	20
Şekil 14. Quirk'in arayüzü (URL-7, 2019). .....	23
Şekil 15. QISKIT'in arayüzü (URL-2, 2019). .....	24
Şekil 16. QISKIT'in ölçüm arayüzü (URL-2, 2019). .....	24
Şekil 17. Ölçüm yapıldıktan sonraki durumu (URL-2, 2019). .....	24
Şekil 18. Not Kapı (X Kapı) Kodu .....	25
Şekil 19. Pauli-X Kapısı ve ölçümü .....	25
Şekil 20a. Devre çalıştırılmadan önce. .....	26
Şekil 21. Sistem çalıştırıldıktan sonra. .....	26
Şekil 22. Grover algoritmasının adımları .....	28
Şekil 23. Grover Algoritması ( $ 11\rangle$ arama) .....	28
Şekil 24. Grover algoritmasının kodu .....	29
Şekil 25. Grover algoritmasının sistemi .....	29
Şekil 26. Girişler. ....	30
Şekil 27. Girişleri Hadamard matrisiyle çarpıldığında. ....	30
Şekil 28. İkinci Adımdaki işlemler. ....	30
Şekil 29. Üçüncü Adım CNOT operatörü .....	31

Şekil 30. Dördüncü Adım.....	32
Şekil 31. Beşinci Adım.....	32
Şekil 32. Altıncı adımdan sonra durum vektörü.....	33
Şekil 33.Yedinci adım sonrasında sistemin durumu .....	34
Şekil 34.Sekizinci adım. ....	34
Şekil 35. Dokuzuncu adımda kubitlerin tersi alınır.....	35
Şekil 36.Onuncu adımda Hadamard kapıları kullanılmaktadır. ....	35
Şekil 37. Grover algoritmasını simulasyonda çalıştırdığımızda.....	35
Şekil 38. IBM Yorktown, 5 kubitlik kuantum bilgisayarında Grover algoritmasının çalıştırılması sonucu.....	36



## TABLolar DİZİNİ

### Sayfa No

Tablo 1. X kapısının doğruluk tablosu: .....	11
Tablo 2. Y kapi doğruluk tablosu: .....	12
Tablo 3. Z kapısının doğruluk tablosu: .....	13
Tablo 4. CNOT kapısının doğruluk tablosu: .....	16
Tablo 5. SWAP kapısının doğruluk tablosu: .....	18
Tablo 6. CCNOT kapısının doğruluk tablosu: .....	19
Tablo 7. CSWAP kapısının doğruluk tablosu: .....	20

## SEMBOLLER DİZİNİ

C	Complex Numbers.
CNOT	Controlled Not Gate.
CCNOT	Controlled Controlled Not Gate.
CSWAP	Controlled Swap Gate.
GPU	Graphical Processing Unit.
H	Hadamard Gate.
HPQC	High-Performance Quantum Computing.
IN	Input.
OUT	Output.
QASM	Quantum Assembly Language.
QCs	Quantum Computers.
QFT	Quantum Fourier Transform.
QISKIT	Quantum Information Science Kit.
QPLs	Quantum Programming Languages.
QUBIT	Quantum Bit.

# 1. GENEL BİLGİLER

## 1.1. Giriş

Kuantum Bilgi işlemleri olarak adlandırılan, bilgisayar mühendisliği ve fizikten türeyen yeni bir bilim dalıdır. Klasik bilgisayar bilimi modelleri ile Bilgi işlem sistemleri arasında kuantum mekaniği yasalarına göre kuantum fenomenini bağlar. Bu fikirler zaman gittikçe geliştirilmiştir. Tam donanımlı fiziki bir kuantum bilgisayar üretilmediği için ve var olan kısıtlı sistemlere erişim yapılamadığından klasik bilgisayarlarda kuantum bilgisayarın simülasyonunu yapılacaktır. 1981 yılında, kuantum mekaniğine dayalı bir hesaplama sisteminin (kuantum bilgisayar) klasik bilgisayarlardan daha fazla güce sahip olabileceğini, ilk ortaya atan Richard Feynman'dır. Bir yıl sonra Uluslararası Teorik Fizik Dergisi'nde kuantum bilgisayarın, kuantum fiziğini yetkin bir şekilde benzetim edebileceğini göstermiştir (Feynman, 1982).

Aynı yıl, Paul Benioff, kuantum mekanik sistemlerinin kuantum Turing makinelerini modelleyebileceğini açıkça göstermiştir (Benioff, 1982).

David Deutsch, evrensel kuantum bilgisayarın bir fiziksel sistemi doğru bir şekilde simüle edebileceğini göstermiştir. Ayrıca evrensel Turing makinesinin bazı paralel hesaplamaları tek bir kayıt cihazında tutamayacağını belirtmiştir (D. Deutsch, 1985).

1989 yılında, Deutsch kuantum devreleri olarak adlandırılan yeni bir kuantum hesaplama modeli tarafından tanıtılmıştır. Klasik hesaplama yapmak için lojik kapılarını uygun şekilde bir araya getirilmelidir. Deutsch, benzer şekilde kuantum hesaplama yapabilmek için kuantum kapılarının bağlanabileceğini kanıtladı. Böylece klasik bilgisayarın hesaplayabileceği her şeyi kuantum devrelerle hesaplanabileceğini duyurdu (D. E. Deutsch, 1989).

1993 yılında, Andrew Chi-Chih Yao, Deutsch'un çalışmalarına kaldığı yerden devam etti. Polinom zamanda bir işlemin kuantum Turing makinesi ile hesaplanabileceğini göstermiştir. Aynı zamanda, polinom büyüklüğünde bir kuantum devresi ile de hesaplanabilir. Bu çalışmalar doğrultusunda araştırmacıların kuantum devrelerine odaklanmalarını sağladı (Yao, 1993).

Zamanla Deutsch'un fikri araştırmacılar tarafından geliştirilmiştir. 1994 yılında, bu konuya, iki problem çözen Peter Shor katkıda bulunmuştur, bunlar: bir tamsayının çarpanlarını bulmak ve ayrık logaritmayı hesaplamak için geliştirilen kuantum

algoritmalarıdır. Shor' un arařtırmaları, kuantum bilgisayarların Turing makinelerinden daha güçlü olduğunu kanıtlamıřtır (Shor, 1994).

Adriano Barenco ve ark. 1995'te sundukları alıřmada, bir qubit kuantum kapılarını ve iki qubit üzerinde kontrol edilen özel OR kapılarını kullanarak kuantum devresini inşa edebilmektedir (Barenco vd., 1995). 1995 yılında, Lov Grover tarafından indislenmemiř veri içinde arama yapma iřlemini kuantum bilgisayarı kullanarak hızlandırılabilirdiğini kanıtlamıřtır (Nielsen ve Chuang, 2002).

2017 yılında Khammassi ve ark. kuantum devrelerinin alıřmasını anlamak için fonksiyonel assembly programlama dilini destekleyen QX simülatörünü tanıtmıřlardır. Simülasyonda özel veri yapılarını kullanarak bellek kullanımını minimize edilebileceğini, göstermiřlerdir (Khammassi vd., 2017).

Sonraki bölümlerde, tezdeki kullanılan bölümleri sunmaktadır, ilk olarak kuantum hesaplama, Programlama dili, hesaplama yöntemi ve aynı zamanda iřlemcilerin evrimini gösterir. 1965'te Gordon Moore devrelerde kullanılan transistörlerin her 18 ayda 2 katına ıkacađını öngörmüřtür. Yaklařık 40 yıl boyunca geçerliliđini koruduđu için Moore yasası olarak isimlendirilmiřtir (Schaller, 1997). Moore yasasına göre birim alanda artan transistör sayısı, transistor boyutlarının küçüleceđi anlamına gelmekteydi. Ancak günümüzde atomik boyutlara eriřildiđi için Moore yasasına uyumlu küçülme gerekleřtirilememektedir.

Atomik boyutta cihazlar üretebilmek için kuantum fiziđinin tanımlarına uygun hareket edilmelidir. Kuantum mekaniđinde tanımlı süperpozisyon ve dolanıklık (entanglement) kavramları anlaşılmalıdır. Kuantum mekaniđi yasalarını kullanarak yeni bir hesaplama modeli (kuantum hesaplama) geliřtirebiliriz.

## **1.2. alıřmanı Amacı ve Önemi**

Bu arařtırmanın amacı, kuantum hesaplamanın temel kavramlarını açıklamak ve kuantum kapılarının ve devrelerin anlaşılmasını sađlamaktır. Uygulama olarak IBM'in geliřtirdiđi ve 5 qubitlik gerek quantum bilgisayarlarla alıřabilen QISKIT kullanılacaktır.

### **1.3. Yüksek performans Hesaplama**

Günümüzde modern bilgisayarlar bir dizi görevi paralel olarak tamamlayabilmektedir. Yazılım geliştiriciler, algoritmaları geliştirirken ve programları oluştururken bilgisayarların sağladığı bu özellikleri göz önünde bulundururlar. Bu şekilde uygulamalarda yüksek performans elde edilir.

Farklı yöntemler arasında seçim yapmak için yüksek performans modelleri kullanmaktadır ve bilgisayarda mümkün olan en iyi işlemi yapıp yapmadığını belirlemek için aksi halde bir zamanlama sonucu fark edebilmektedir (Eijkhout, 2013). Zamanla, iki sebepten dolayı yüksek performans hesaplamanın gereği anlaşılmıştır. Birincisi, yüksek performans seviyelerine ulaşmak için uygun teknoloji altyapısı bulunmaktadır. Örneğin süper bilgisayar milyonlarca işlemcinin harmoni içinde çalışabilen hesaplama merkezleridir. İkinci ve en önemli sebep ise, kişisel ve ticari bilgisayar pazarının kullanıcının artan performans ihtiyacını karşılayabilmektedir (Dowd ve Severance, 2010). Performans ihtiyacını karşılamak için farklı amaçlara hizmet eden hızlandırıcılar geliştirilmiştir. Bunlar arasında grafik kartları (GPU), FPGA'lar (Field Programmable Gate Arrays) en önemli olanlarıdır. FPGA'lar aslında donanım geliştirme için tasarlanan test devreleridir. Ancak günümüzde onlarca işlemci çekirdeği bir FPGA içine yerleştirilebilmekte ve binlerce FPGA bir araya getirilerek süper bilgisayarlar oluşturulabilmektedir.

### **1.4. Kuantum Mekaniğin Temelleri**

#### **1.4.1. Kuantum Fizik**

Bu çalışmada yapılan kuantum hesaplama simülasyonunun anlaşılabilmesi için bu bölümde kuantum mekaniğinin temel bilgileri açıklanmıştır. Newton tarafından tanımlanan klasik fizikte, cisimleri ve onların hareketleri tanımlanmaktadır. Cisimler belirtilen pozisyon ve faaliyete sahip ayrı bir varlıktırlar. Maxwell yasaları ise manyetik alanları ve elektrik akımından kaynaklanan elektromanyetik dalgaları açıklamaktadır. Kısacası parçacığın hareketi ve dalga özellikleri birbirinden bağımsız yasalar altında tanımlanmıştır.

Max Planck, bir sabiti (Planck) değere bağlı olarak "siyah cisim radyasyon teorisi" ni 1900 yılında duyurdu.



Planck'in teorisinde verdiği denklem şu şekildedir:

$$E = h\nu, \text{ burada}$$

$E \equiv$  fotonun enerjisi (kuantum enerjisi)

$\nu \equiv$  Planck sabiti

$h \equiv$  ışığın frekansı (Phillips, 2013).

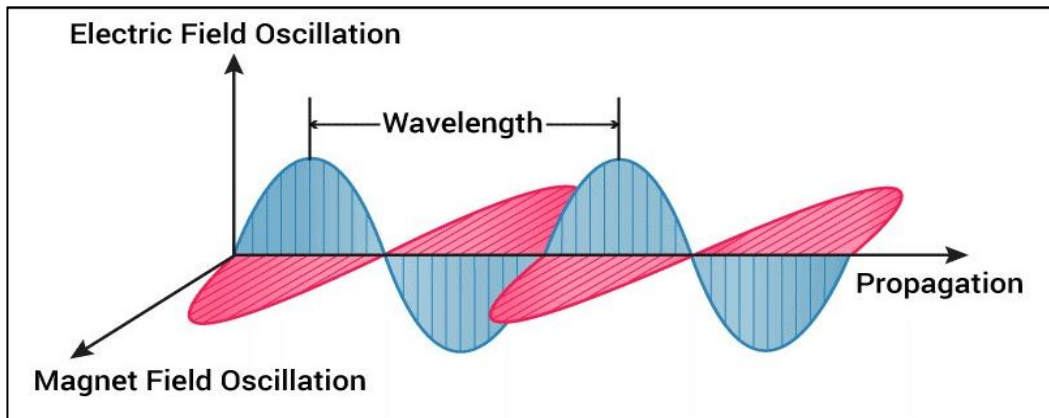
Planck teorisiyle cismin hareketi ve dalga özelliği bir çatı altında açıklanabilmektedir. Bu duyurudan sonra bilimsel çalışmaların yoğunluğu bu teoriye yoğunlaşmış ve klasik fizik zayıflamaya başlamıştır. Temelleri 20. Yüzyılın başında atılan Kuantum Fiziği, parçacıkların atom altı seviyedeki davranışlarını açıklayan bilim dalı olarak günümüze kadar geçerliliğini korumaktadır.

#### 1.4.2. Fotonlar ve Işık

Foton elektromanyetik radyasyonun parçacık benzeri kuantlarıdır (Phillips, 2013).

#### 1.4.3. Işık

Bir foton akımıdır, ancak ışık aynı zamanda elektromanyetik bir dalgadır, elektrik ve manyetik salınımların dalganın yayılma yönüne dik açılarda olduğu anlamına gelir ve şekil 1'de gösterilmiştir.



Şekil 1. Dik açılarda (URL-8, 2019).

#### 1.4.4. Süperpozisyon

Klasik fizikte cisimler yalnızca bir durumda bulunabilir, ölçüm sonucunda da bulunduğu pozisyon tespit edilir. Bir kuantum sistemin durumunu bulunabileceği durumların karışımı olarak ifade edilir. Eğer bir sistem A ve B durumlarında olabilirse, bu iki durumun bir “karışımı” şeklinde  $aA+bB$  gösterilebilir. Burada a, b durumlarda bulunma olasılıklarını veren karmaşık sayılardır. Kuantum sistem ölçüm sonucunda bozulur ve sistemin durumu klasik durumlardan birisine çöker. Kısacası ölçüm sonucunda sistem  $a^2$  olasılıkla A durumuna  $b^2$  olasılıkla B durumuna çöker (Patvardhan, 2019). Sistem uniform olduğu için normalize şartını sağlamalıdır. Normalize şartı  $a^2+b^2=1$  eşitliği ile belirtilir.

#### 1.4.5. Dolanıklık (Entanglement)

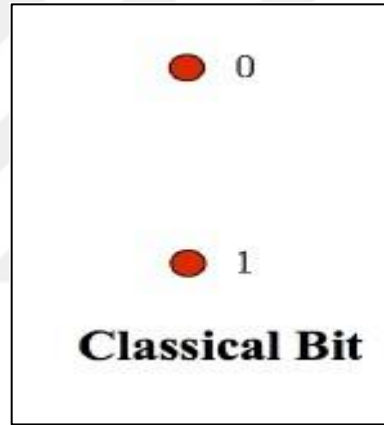
Çok parçalı bir sistemde dolanıklık, sistemin bir kısmında yapılan işlemin diğer kısımlarına da etkilemesi anlamına gelmektedir. Örneğin Shor algoritmasında dolanık olarak hazırlanan iki registerdan ikincisinde yapılan ölçüm sonucuna göre birinci register belli değerlere çöker.

## 2. KUANTUM HESAPLAMAYA GİRİŞ

Kuantum hesaplama, klasik bilgisayar bilimi ve kuantum mekaniğinin sonuçlarını birleştirmektedir. Bu bölümde, kuantum hesaplamının temelleri genel bir şekilde sunulmuştur.

### 2.1. Kuantum Bit

Klasik hesaplama alanında, en küçük bilgi biriminin - bit - sadece 0 veya 1 değerlerini alabildiğini biliyoruz. Ayrık iki seviyeli bir sistemdir, sistemin değeri tam olarak hangi seviyede olduğunu göstermektedir.



Şekil 2. Klasik bit.

Kuantum hesaplama alanında, en küçük bilgi birimi Kubit'tir. Kubit KUantum bit kelimelerinden türetilmiştir. Kubit iki boyutlu bir Hilbert uzayı ile temsil edilir. Bu nedenle iki ortonormal vektör ile ifade edilir. Herhangi bir hesaplama yapabilmek kubit'i tanımlamak için bir çift ortonormal vektör seçeriz:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ ve } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

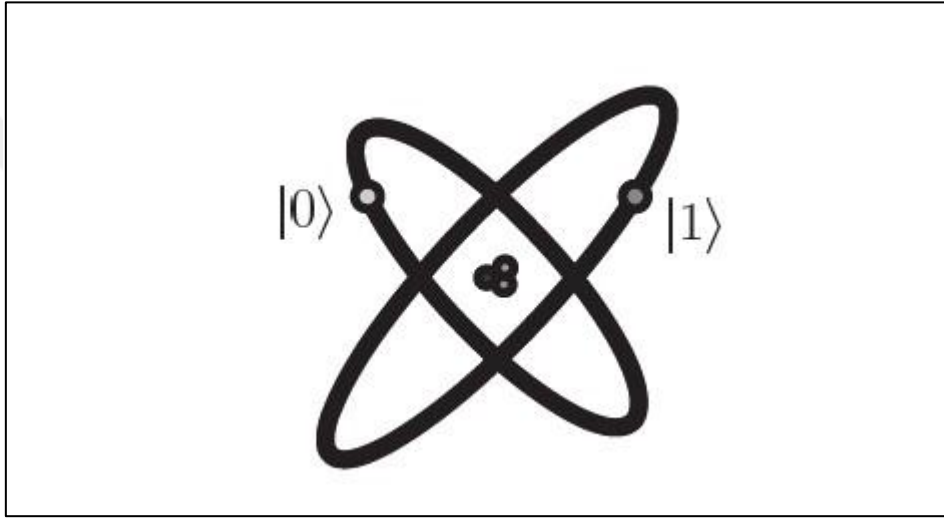
Örneğin bu vektörler, standart taban (baz) vektörlerdir.

Klasik hesaplamada lojik durumlar gibi, Kubit,  $|0\rangle$  ve  $|1\rangle$  olarak adlandırdığımız iki temel durumdan birinde olabilir veya süperpozisyon durumunda olabilir:

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

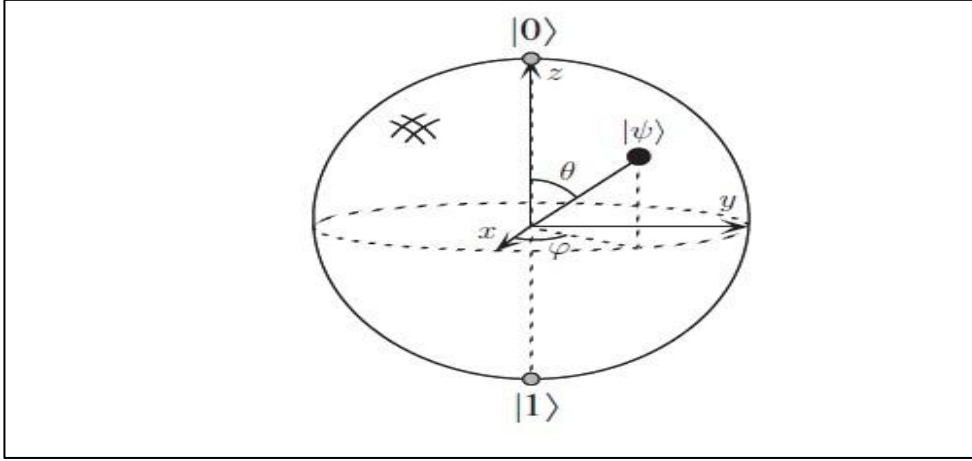
Burada  $\alpha, \beta \in C$  (Karmaşık sayılar) (Mlnařik, 2007)

Bu ifadede ' $| \rangle$ ' sembolü Dirac notasyonu olarak adlandırılmıştır ve Dirac bu ifadeye "ket" vektörü olarak ifade etmiştir. Ket vektörü bir sütun vektördür. Genelde ket 0 ( $|0 \rangle$ ) vektörü sistemin temel durumunu,  $|1 \rangle$  sistemin uyarılmış durumunu temsil eder. Bu sistem Şekil 3'te gösterilmiştir. Bu kubit ölçtüğümüzde,  $|\alpha|^2$  olasılıkla 0 sonucunu alırız, veya  $|\beta|^2$  olasılıkla 1 sonucunu okuruz. Burada normalizasyon şartı gereği  $|\alpha|^2 + |\beta|^2 = 1$  çünkü olasılıklar toplamı 1 olmalıdır (Nielsen ve Chuang, 2002).



Şekil 3. Kubit, bir atomdaki iki elektronik seviye tarafından temsil edilir (Nielsen ve Chuang, 2002).

Şekil 4'teki küreye Bloch küresi denir; Tek bir kubitin durumunu görsel bir şekilde sunar. Genellikle kuantum hesaplama ve kuantum bilgiyi tanımlamada kullanılan güzel bir araçtır (Nielsen ve Chuang, 2002).



Şekil 4. Bir kubitin Bloch küre üzerinde gösterimi (Nielsen ve Chuang, 2002).

## 2.2. Çok Kubitli Sistemler: Kuantum Registerlar

Farz edelimki bir sistemde iki kubit var. Bunlar iki klasik bit olsaydı, dört olası durumdan 00, 01, 10 ve 11 birisiyle ifade edilirdi. Ancak kuantum sistem dört durumun karışımı (süperpozisyonu) olarak ifade edilebilir. Kısacası  $|00\rangle$ ,  $|01\rangle$ ,  $|10\rangle$ ,  $|11\rangle$  bu dört durumun süperpozisyonları 2-kubitli sistemi tanımlayabilir.

Çok kubitli bir sistemi kubitlerin bileşimi olarak yazabiliriz. Her bir kubiti temsil eden vektörü tensör çarpımı yaparak elde edebiliriz. A ve B iki boyutlu sütun vektörler olsun, tensör çarpımı  $a \otimes b$  ile bileşik sistemi ifade edebiliriz.

$$a \otimes b = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \otimes \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 b_1 \\ a_2 b_1 \\ a_1 b_2 \\ a_2 b_2 \end{pmatrix}$$

Şimdi ise baz vektörlerin tensörlerini elde edelim:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Demek ki :

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$|01\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|10\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$|11\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Bu bilgiler ışığında iki kubit bir sistemin durumunu aşağıdaki şekilde verebiliriz:

$$|\Psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \quad (\text{Akama, 2015}).$$



### 3. KUANTUM KAPILAR

Kubitler üzerinde işlem yapabilmek için temel kuantum kapılar tanımlanmıştır. Bu kapılar kubitler üzerinde basit üniter işlemleri yapmamızı sağlar (D. Deutsch, 1985; D. E. Deutsch, 1989). Kuantum kapılar klasik sistemdeki lojik kapılar gibi Kuantum devrelerin en temel bileşenleridir. Kuantum kapıları tek kubit kapılar ve çok kubitli kapılar olmak üzere iki kısma ayrılabiliriz.

#### 3.1. Tek Kubit Kapılar

Kuantum hesaplamının temel işlemlerini yapabilmek için bir kubitli kapılar tanımlanmıştır. Bu kapıların tamamı aldığı bir vektörü üniter olarak dönüştürerek başka bir vektör üretir. Burada, KISQIT devre tasarım platformunda tanımlı olan ve Quantum Assembly dilinde komut karşılığı olan tüm kapılar işlenecektir (URL-2, 2019).

##### 3.1.1. Pauli Kapıları

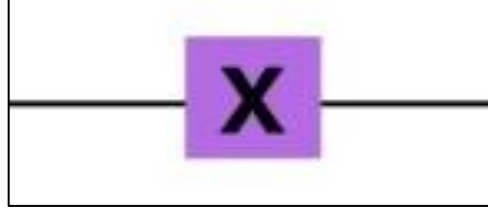
En basit kuantum kapılar Pauli kapılarıdır: Bunların X, Y ve Z olarak isimlendirilmişlerdir. X, Y ve Z kapıları bir kubiti sırasıyla Bloch küresinde x, y ve z eksenleri etrafında yarı bir dönüşünü gerçekleştirmektedir. Bu nedenle klasik NOT kapısına benzerdirler. Özellikle, X kapısının kubit üzerindeki işlemi oldukça açıktır (URL-2, 2019).

###### 3.1.1.1. Not Kapı ( X Kapı )

X kapısı, yalnızca verilen kubit değerini tersine çeviren bir-girişli bir-çıkışlı kapıdır. Pauli X kapısının matris gösterimi:

$$X \equiv \sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Bu kapının devre sembolü şekil 5'de gösterilmiştir.



Şekil 5.X kapısının Grafik sembolü (Hayes, 2003).

Pauli X kapısı baz durumları birbirine dönüştürür. Yani ket 0 girişini ket 1'e, ket 1'i ket 0'a dönüştürür.

$$X|0\rangle = |1\rangle,$$

$$X|1\rangle = |0\rangle$$

X kapısının doğruluk tablosu Tablo 1'de verilmiştir.

Tablo 1. X kapısının doğruluk tablosu:

A	X(IN)	(IN) <sup>t</sup>
$ 1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$
$ 0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$
$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \times \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$	$\begin{pmatrix} \beta \\ \alpha \end{pmatrix}$

### 3.1.1.2. Pauli Y Kapısı

Pauli X kapısına benzer şekilde bu kapı da Bloch Küresi (Bloch Sphere) üzerinde döndürme işlemi yapmaktadır. Ancak bir önceki kapıdan farklı olarak Y ekseninde döndürme işlemi yapılır (URL-5, 2010).

Y kapısının matrisi:

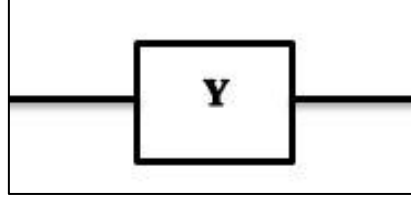
$$Y \equiv \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

Pauli Y kapısı aşağıdaki gibi basitleştirilebilir:



$$Y = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Y kapısının sembolü şekil 6 ile belirtilmiştir:



Şekil 6.Y kapı Grafik sembolü.

Tablo 2.Y kapı doğruluk tablosu:

<b>a</b>	<b>Y(IN)</b>	<b>(IN)<sup>t</sup></b>
$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} -1 \\ 0 \end{pmatrix}$
$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$
$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \times \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$	$\begin{pmatrix} -\beta i \\ \alpha i \end{pmatrix}$

### 3.1.1.3. . Z kapı

Daha önce de belirtildiği gibi Pauli X ve Y kapılarına benzer bir işlevi vardır. Bir kubit'i Bloch Küresinde Z ekseninde döndürme işlemi yapar (URL-5, 2010).

Z kapısının matrisi:

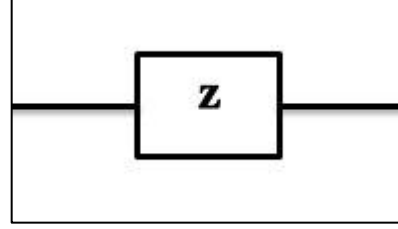
$$Z \equiv \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Z kapı  $|+\rangle$  ve  $|-\rangle$  durumları üzerinde X gibi benzer bir etkiye sahiptir:

$$Z|+\rangle = |-\rangle$$

$$Z|-\rangle = |+\rangle$$

Pauli Z kapısının devre sembolü şekil 7'de verilmiştir:



Şekil 7. Z kapısının sembolü.

Tablo 3. Z kapısının doğruluk tablosu:

A	Z(IN)	(IN) <sup>t</sup>
$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ -1 \end{pmatrix}$
$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$
$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \times \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$	$\begin{pmatrix} \alpha \\ -\beta \end{pmatrix}$

### 3.1.2. Hadamard Kapısı (H)

Pauli kapıları gibi, Hadamard kapısı da Bloch küresi üzerinde yarı dönme hareketini tanımlar. Aradaki fark, x ve z eksenlerinin ortasından kesen bir eksen etrafında dönme sağlar. Bu, z eksenini boyunca işaret eden ve x boyunca işaret edenlere dönen durumların etkisini verir (URL-2, 2019). Birçok kuantum algoritması, Hadamard kapısını bir registerdaki her bir qubit'e uygulayarak başlar.

Hadamard H kapısının matrisi:

$$H \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Buna karşılık gelen Dirac notasyonu aşağıda verilmiştir.

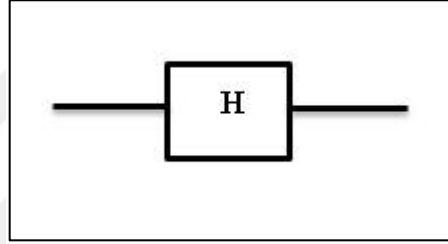
$$H = \frac{|0\rangle + |1\rangle}{\sqrt{2}} |0\rangle + \frac{|0\rangle - |1\rangle}{\sqrt{2}} |1\rangle$$

$|0\rangle$  değerini  
 $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$  olarak

$|1\rangle$  değerini

$\frac{|0\rangle - |1\rangle}{\sqrt{2}}$  olarak dönüştürür

H kapısının devre sembolü şekil 8'de gösterilmiştir:



Şekil 8. Hadamard kapısının sembolü.

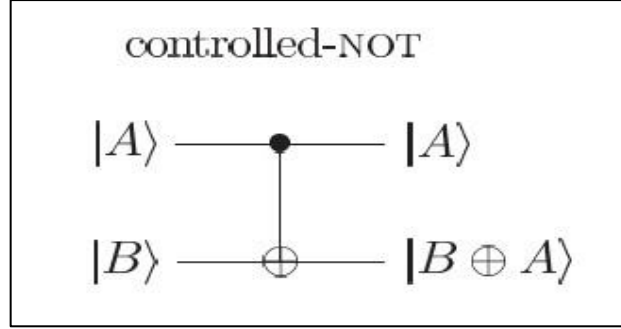
### 3.2. Çok Kubitli Kapıları

Bu kapılar birden fazla kubitli işleyen kuantum sistemlerdir. Bir kuantum sistem terslenebilir (reversible) olduğundan giriş sayısı kadar çıkış kubitli vardır. Bu durum kapılar için de geçerlidir. Yani bir kapının iki girişi varsa iki çıkışı, 3 girişi varsa 3 çıkışı bulunur. Örnek olarak iki kubitli işleyen CNOT kapısının devre gösterimi Şekil 9'da gösterilmiştir (Nielsen ve Chuang, 2002).

Bu şekilde ikin kubitli bir sistem dört boyutlu sütun vektörlerle temsil edilmektedir:

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \equiv \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, |01\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|10\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, |11\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$



Şekil 9. Çok kubitli kapı örneği (Nielsen ve Chuang, 2002).

### 3.2.1. Kontrollü NOT apısı

Kontrollü NOT, terslenebilir bir kapıdır. 2 girişli kubit ve 2 çıkış kubit bulunur. Kubitlerden biri, diğerini kontrol ettiği için bu ismi almıştır. Kısacası CNOT kapısı olarak adlandırılır. CNOT kapısının matris gösterimi aşağıda verilmiştir.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

CNOT kapısı, ilk kubit “ket 1” durumunda ise ikinci kubit’in tersi alınır. İlk kubit, ikincisi için kontrol girişi olarak işlev görmektedir. Doğruluk tablosu Tablo 4’da gösterilmiştir. Grafik sembolü Şekil 10’de gösterilmektedir. (Nielsen ve Chuang, 2002).

$$CNOT |xy\rangle \rightarrow |xy \oplus x\rangle$$

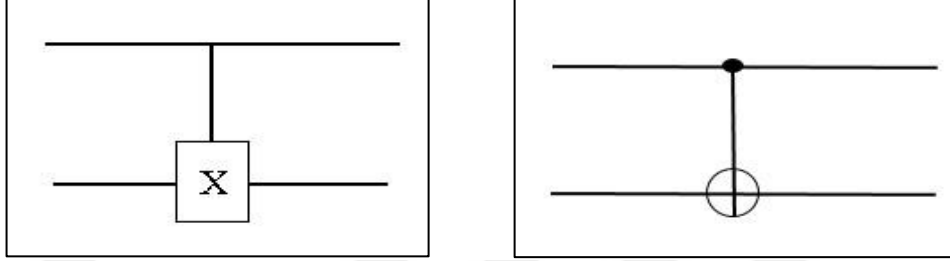
Demek ki ilk kubit 1 olunca, ikinci kubit tersine çevrilir.

Örnek :

$$CNOT |11\rangle \rightarrow |1\ 1 \oplus 1\rangle \rightarrow |10\rangle$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Şekil 10'da CNOT kapısının devre sembolleri verilmiştir.



Şekil 10.CNOT kapı Grafik sembolü.

CNOT kapısının doğruluk tablosunu (bkz. Tablo 4) incelediğimizde kontrol kubitini olarak adlandırılan kubitin değişime uymadığını, diğer kubitin ise yalnızca birinci kubit 1 ise değiştiğini görebiliriz.

Tablo 4. CNOT kapısının doğruluk tablosu:

IN	CNOT(IN)	OUT
$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$
$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$
$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$
$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$

### 3.2.2. Swap Kapısı

2 girişi / 2 çıkış olan bir kuantum kapısıdır. SWAP, girişindeki kubitleri yer değiştirir (Williams, 2010).

SWAP kapısının matrisi:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

**SWAP**  $|xy\rangle \rightarrow |yx\rangle$

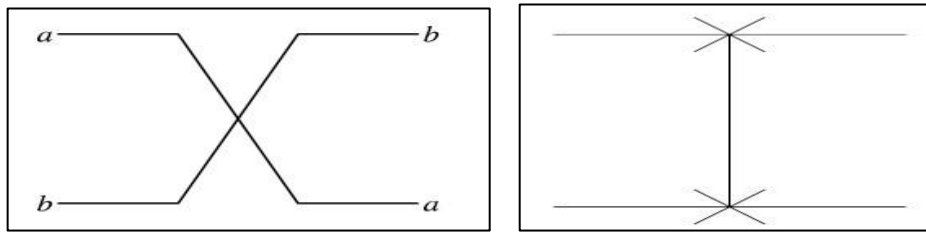
Örnek:

SWAP  $|00\rangle = |00\rangle$ , SWAP  $|01\rangle = |10\rangle$ , SWAP  $|10\rangle = |01\rangle$ , SWAP  $|11\rangle = |11\rangle$ .

Bu kapıda  $|01\rangle$   $|10\rangle$ 'a ve  $|10\rangle$   $|01\rangle$ 'e değiştirilir ve  $|00\rangle$  ve  $|11\rangle$  hiçbir değişiklik olmaz (Ben-Aryeh, 2009).

SWAP kapısının sembolü şekil 11' de gösterilmektedir. Doğruluk tablosu Tablo 5'de verilmiştir.

SWAP kapı Grafik sembolü:



Şekil 11.SWAP kapısının Grafik sembolü (Williams, 2010).

Tablo 5. SWAP kapısının doğruluk tablosu:

IN	SWAP(IN)	OUT
$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$
$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$
$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$
$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$

### 3.2.3. . Toffoli Kapı (CCNOT)

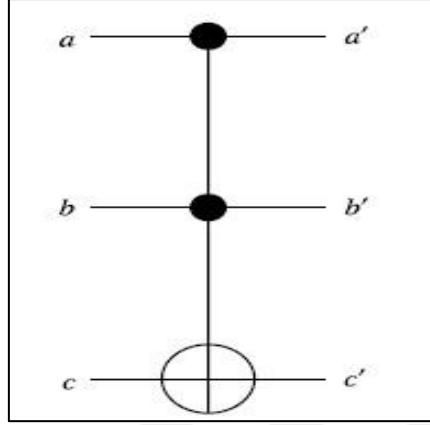
3 girişi, 3 çıkışı olan bir kuantum kapıdır. CNOT kapısına bir kontrol girişi daha elde edilerek oluşturulduğu için kontrollü-kontrollü-NOT kapısı olarak da bilinir (CCNOT) Çünkü üçüncü girişin terslenmesi için (NOT) ilk iki girişin 1 olması gerekmektedir. Diğer bir deyişle, ilk iki giriş değerleri, üçüncü kubitin terslenip terslenmeyeceğini kontrol eder. (Williams, 2010).

CCNOT kapısının matrisi:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$CCNOT|abc \rangle = |ab \oplus c \rangle$$

CCNOT kapısının sembolü şekil 12'de gösterilmektedir. Doğruluk tablosu Tablo 6'da verilmiştir.



Şekil 12. CCNOT kapısının sembolü.

Tablo 6. CCNOT kapısının doğruluk tablosu:

IN <sub>1</sub>	IN <sub>2</sub>	IN <sub>3</sub>	OUT <sub>1</sub>	OUT <sub>2</sub>	OUT <sub>3</sub>
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Koyu olan kısımlar 1.ve 2. Kontrol girişlerine göre çıkışın değiştiği kısımlardır.

### 3.2.4. . Fredkin Kapısı (CSWAP)

Fredkin kapısı da 3 girişi, 3 çıkışı olan bir kapıdır. Aynı zamanda kontrollü SWAP kapısı olarak da bilinir (CSWAP). Birinci giriş 1 olduğunda ikinci ve üçüncü girişler yer değiştirir. Diğer bir deyişle, birinci kubit değeri, ikinci ve üçüncü kubitlerin yer değiştirmesine karar verir (Williams, 2010).

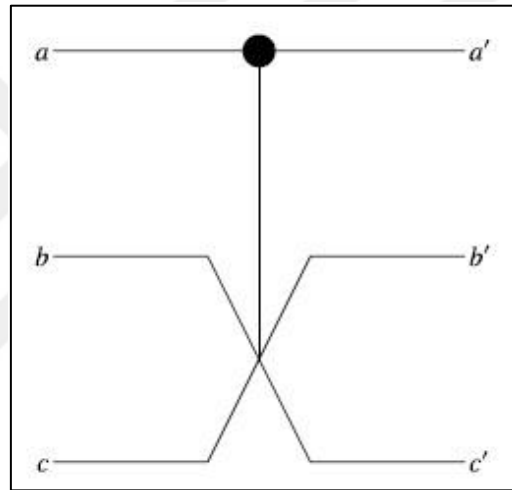


CSWAP kapısının matrisi:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{CSWAP}|abc\rangle = |a\oplus cb\rangle$$

CSWAP kapı Grafik sembolü Şekil 13'de gösterilmektedir. Doğruluk tablosu Tablo 7'da gösterilmiştir.



Şekil 13. CSWAP kapısının sembolü.

Tablo 7. CSWAP kapısının doğruluk tablosu:

IN <sub>1</sub>	IN <sub>2</sub>	IN <sub>3</sub>	OUT <sub>1</sub>	OUT <sub>2</sub>	OUT <sub>3</sub>
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

Koyu olan kısımlar birinci kubit 1 olduğunda diğerlerinin değerlerinin yerdeğiştirdiğini göstermektedir.



## 4. KUANTUM PROGRAMLAMA DİLLERİ

OPENQASM 2.0, bir kuantum devresini tanımlamak için düşük seviyeli bir dildir. Ancak daha çok kuantum devrelerin icrası ve simülasyonu ile ilgili daha fazla özellik tanıtmaktadır.

OPENQASM programının ilk (yorumsuz) satırı OPENQASM ile başlamaktadır. İfadeler noktalı virgüllerle ayrılır. Boşluklar yok sayılmaktadır. Dil büyük / küçük harf ayırımına duyarlıdır. Yorum satırları c/c++'daki gibidir.

Bu bölümde, OPENQASM dilinin sözdizimi ve kuantum kodunun anlamı açıklanacaktır. Okuyucunun bunu anlamasına yardımcı olmak için devre örnekleri kullanılarak anlatılacaktır (Cross vd., 2017; Khammassi, 2016; URL-4, 2016).

### 4.1. Gösterimler

1. OPENQASM, QISKIT ile bütünleşik çalışmaktadır.
2. Kod editörü önceden tanımlanmış anahtar kelimeleri renkli olarak belirtir, örneğin: "qubits", "" **measure** ", " **cnot** "...

### 4.2. Sözdizimi

OPENQASM dili bir kuantum Assembly dilidir. Bu nedenle komutlar kuantum operatörlerin çalışmasını ifade etmektedir. Aşağıda verilen kod parçasında **q** bir kubit registerdir. Bu registerin ilk kubit Pauli X kapısıyla işlenmektedir. İkinci Satırda ise işlenen bu kubit ölçüm yapılarak sonuç **c** klasik registerin ilk bitine yazılmaktadır.

```
x q [0]; // pauli x kubit 0'de  
measure q [0] -> c [0]; // ilk kubit ölçmesi
```

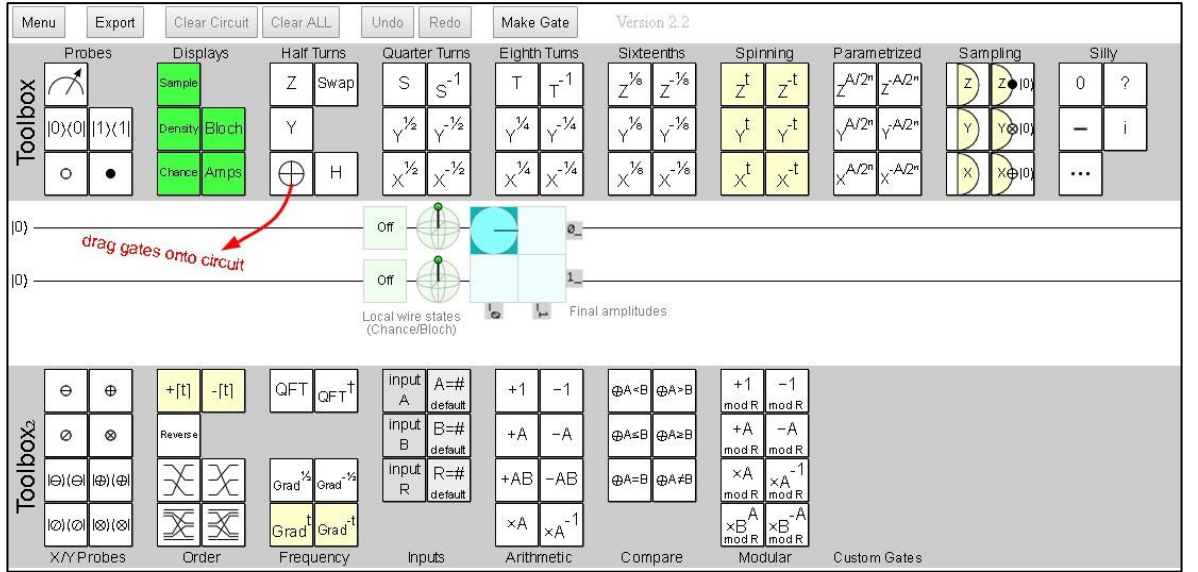
Bilindiği gibi dünyada toplu kullanıma açık bir şekilde sunulan kuantum bilgisayarlar çok sınırlıdır. Bunların arasında IBM tarafından kullanıma sunulan iki adet 5 kubitlik sistemler bulunmaktadır. Aynı şekilde IBM tarafından geliştirilen QISKIT sistemi ile bu fiziksel sistemlerde test kodları denenebilmektedir. Bilindiği gibi 5 kubit bir

sistemde sağlıklı bir kuantum sistem sentezlenemez. Bu nedenle QU Touch şirketinin bir ürünü olan QX gibi bazı simülatör kullanarak kuantum kapılarını simüle edebiliriz. Böylece klasik bilgisayarlarda benzetmeye çalışmaktadır.

### 4.3. QUIRK Kuantum Simulasyon Sistemi

Quirk, açık kaynaklı bir yazılımdır. Kaynak kodu izin verilen bir apache lisansı altında bulunmaktadır. Küçük kuantum devrelerini Tasarım ve Kontrol edebilmek için harika bir sürükle ve bırak kuantum devre simülatörüdür (URL-6, 2019).

Resim:

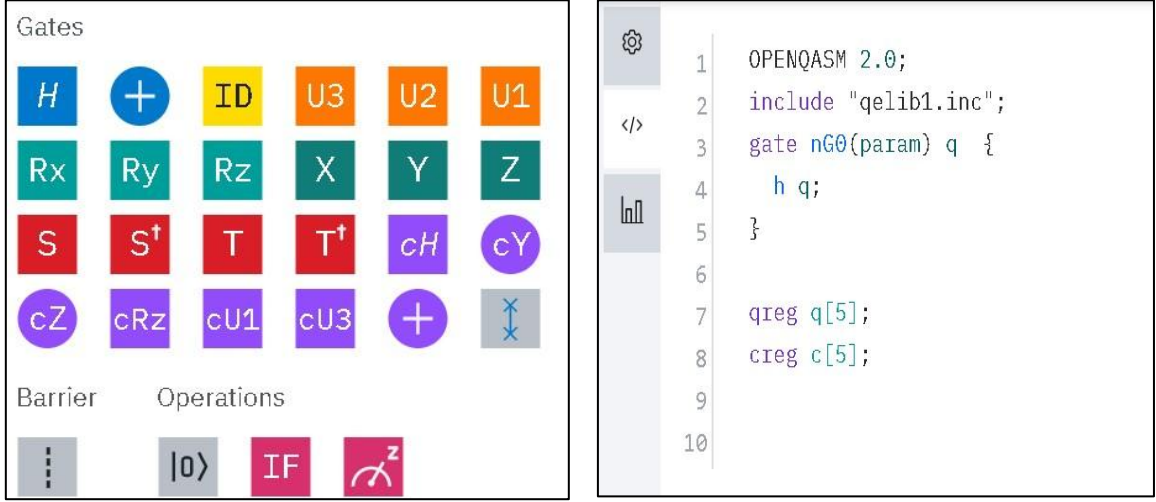


Şekil 14. Quirk'in arayüzü (URL-7, 2019).

### 4.4. Kuantum Bilgi Bilim Kiti (QISKIT)

QISKIT açık kaynak kodlu bir online bir simülatördür. Yerel bir makinede veya çevrimiçi olarak çalışan çok yönlü bir sistemdir. Ayrıca bu simülatör bahsedilen iki tane 5 kubit sistemde testler yapmamızı sağlar. QISKIT'ta bir kod editörü (QASM) ve grafiksel bir kullanıcı arayüzü bulunmaktadır (URL-2, 2019).

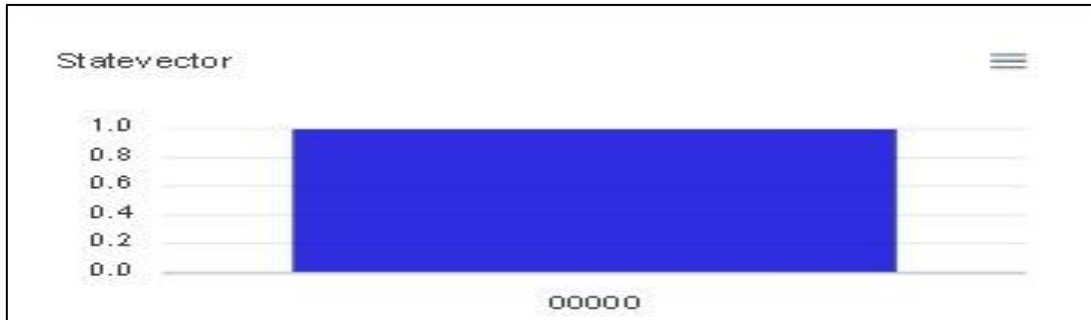
QISKIT'te hem sürükle bırak yöntemiyle kuantum devrenin şeması oluşturulabilir. Hem de OPENQASM kullanılarak doğrudan kod yazılabilir:



Şekil 15. QISKIT'in arayüzü (URL-2, 2019).



Şekil 16. QISKIT'in ölçüm arayüzü (URL-2, 2019).



Şekil 17. Ölçüm yapıldıktan sonraki durumu (URL-2, 2019).

#### 4.4.1. Pauli-X Kapısının ( kuantum NOT) Kodu

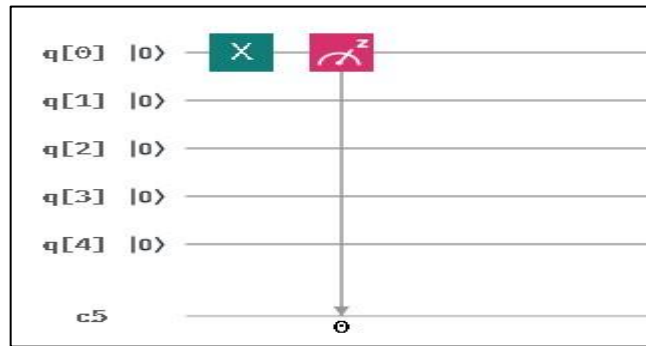
Pauli X kapısının OPENQASM dilinde kodu aşağıda verilmiştir. Kütüphane eklendikten sonra 5 kubitlik bir kuantum register tanımlanmıştır. Ölçüm sonuçlarının saklanması için de 5 bitlik klasik register tanımlanmıştır.

```
OPENQASM 2.0;
include "qelib1.inc";
gate nG0 (param) q {h q;}
qreg q [5]; //5 kubitlik bir kuantum register tanımlanmıştır.
creg c [5]; //5 bitlik klasik register tanımlanmıştır
x q [0]; //pauli x qubit 0'de
measure q [0] -> c [0]; // ilk qubit ölçmesi ve saklaması
```

Şekil 18. Not Kapı (X Kapı) Kodu.

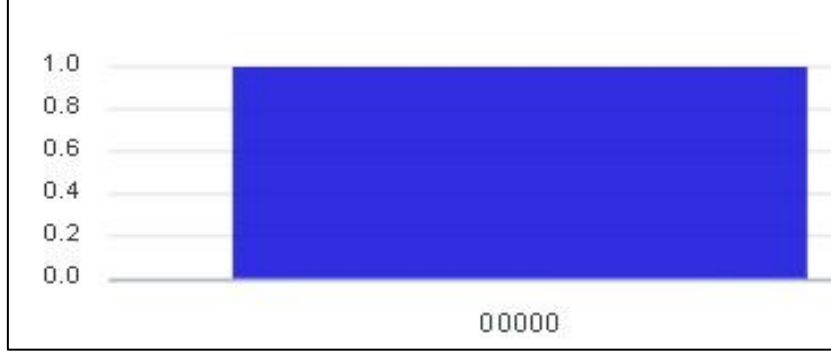
Bu kuantum sistemi QISKIT arayüzünden görsel bir şekilde hazırlayabiliriz.

Şekil 18, Pauli-X kapısının ket 0 kubitini terslediğini ifade eden kuantum sistemi göstermektedir

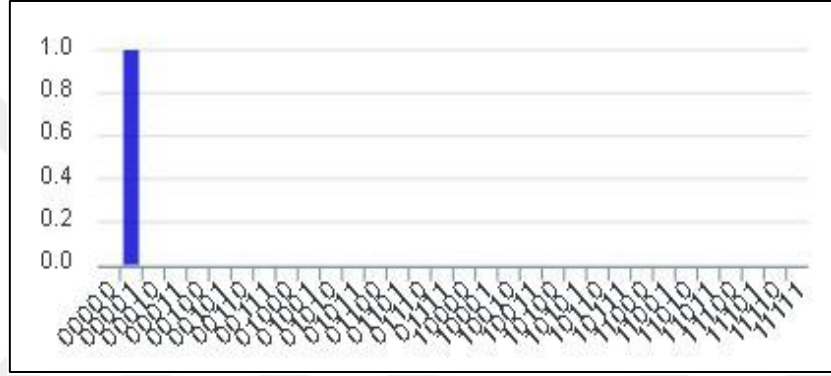


Şekil 19. Pauli-X Kapısı ve ölçümü.

Şekil 19a'da sistem çalıştırılmadan önce  $|00000\rangle$  durumuna set edildiğini göstermektedir. Pauli-X kapısı ilk kubit tersledikten sonra sistemin yeni durumu  $|00001\rangle$  olarak ölçülmüştür (bkz. Şekil 20b).



Şekil 20a. Devre çalıştırılmadan önce.



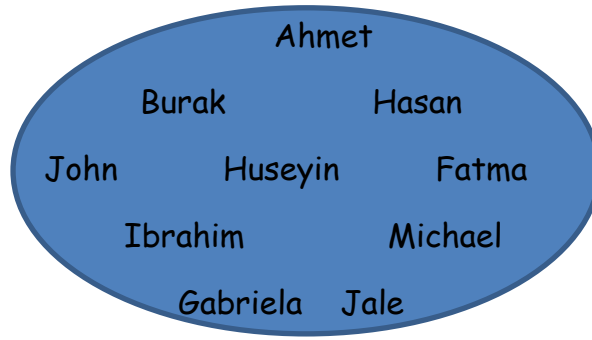
Şekil 21. Sistem çalıştırdıktan sonra.

#### 4.4.2. Grover Arama Algoritması

Kuantum Veritabanı Araması, 1996 yılında Lov K. Grover tarafından geliştirilmiştir. Kuantum Bilgisayarlar için bir arama algoritmasıdır. Dizine eklenmemiş verileri aramak için etkili bir algoritmadır.

Örnek: Tüm Kongre Kütüphanesinde arama yapmak için veya indekslenmemiş bir veritabanında bir kişinin adının sorgulanması (Grover L.K, 1996).

Bu işlem klasik bilgisayarlarda yıllarca sürebilir. Ancak, Kuantum Bilgisayarlarda bu arama işlemi  $1/2$  saniyelik bir işlem olacaktır.

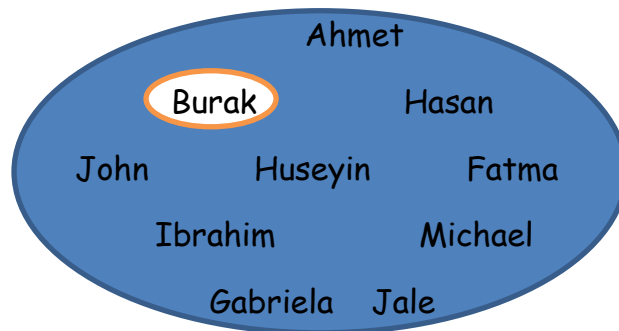


ara: **Ibrahim?**

- Linear arama  $n/2$  adımlarını gerektirir.
- Grover'in algoritması bulmak için  $\sqrt{n}$  adımlarını gerektirir.

#### 4.4.2.1. Brute Force Arama

- Tek tek karşılaştırma yapılır.
- Bulunması  $n$  elemanlı bir küme için...
- $S = \{\text{Ahmet}, \dots\}$ ,  $n = |S|$ ,
- Arama fonksiyonu
- Search(isim)
- $O(\text{search}()) = n$

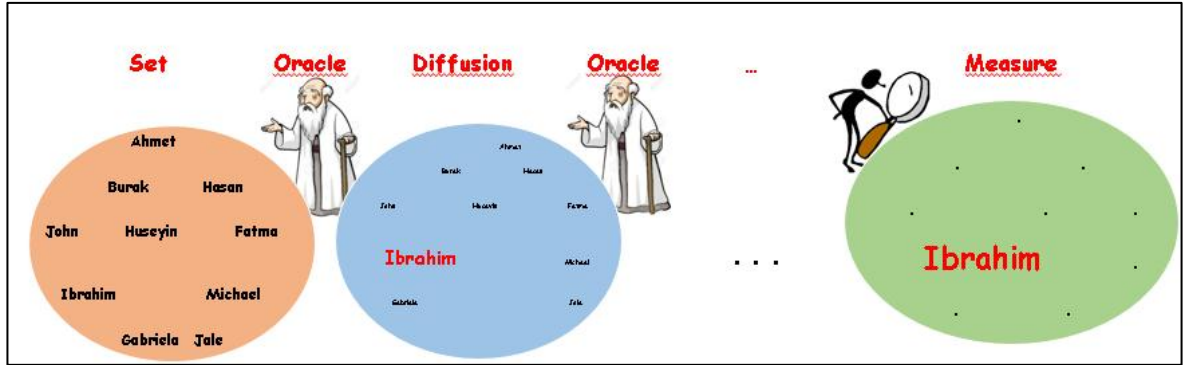


Burak? = **Ibrahim**

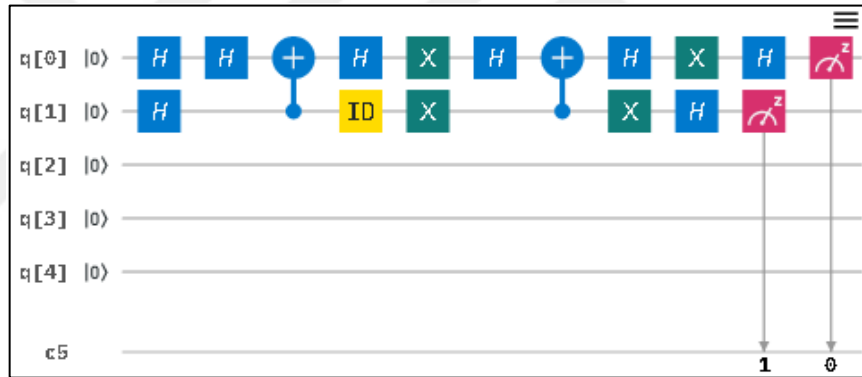


#### 4.4.2.2. Grover Algoritmasının Adımları

Grover algoritmasının adımları aşağıdaki şekilde verilmiştir.



Şekil 22. Grover algoritmasının adımları.



Şekil 23. Grover Algoritması ( $|11\rangle$  arama).

```

OPENQASM 2.0;
include "qelib1.inc";

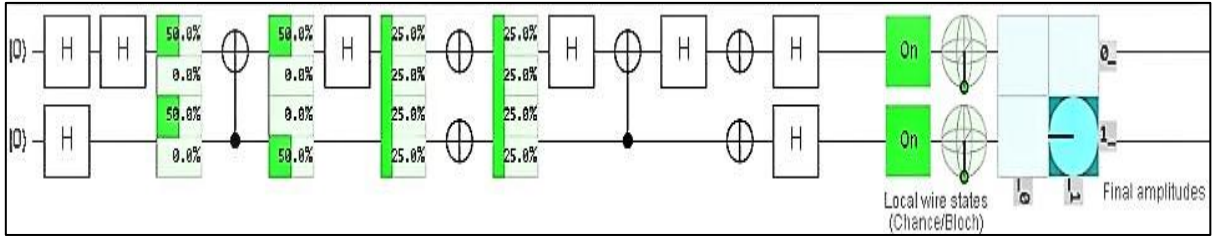
qreg q[5]; //5 kubitlik bir kuantum register tanımlanmıştır.
creg c[5]; //5 bitlik klasik register tanımlanmıştır.

h q[0]; //hadamard qubit 0'de.
h q[1]; //hadamard qubit 1'de.
h q[0]; //hadamard qubit 0'de.
cx q[1],q[0]; //cnot qubit 1 kontrol eden qubit , qubit 0 hedef olan qubit.
h q[0]; //hadamard qubit 0'de.
id q[1];
x q[0]; //not qubit 0'de.
x q[1]; //not qubit 1'de.
h q[0]; //hadamard qubit 0'de.
cx q[1], q[0]; //cnot qubit 1 kontrol eden qubit , qubit 0 hedef olan qubit.
h q[0]; //hadamard qubit 0'de.
x q[1]; //not qubit 1'de.
x q[0]; //not qubit 0'de.
h q[1]; //hadamard qubit 1'de.
h q[0]; //hadamard qubit 0'de.
measure q[1] -> c[1]; // ikinci qubit ölçmesi ve saklaması.
measure q[0] -> c[0]; // ilk qubit ölçmesi ve saklaması.

```

Şekil 24.Grover algoritmasının kodu.

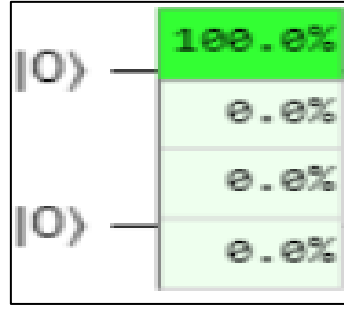
Grover Algoritmasının çalışmasını simülasyon üzerinde açıklayalım:



Şekil 25. Grover algoritmasının sistemi.

- İlk adım

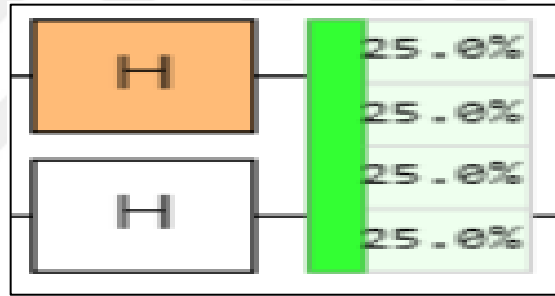
Giriş  $|00\rangle$  dir ve vektör gösterimi ise  $\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$



Şekil 26. Girişler.

2 Hadamard matrisiyle çarparak çıkışı aşağıdaki gibi olacaktır

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} * \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$



Şekil 27. Girişleri Hadamard matrisiyle çarpıldığında.

- İkinci Adım: Bu adımda kubit 0 Hadamard kapısında işlenecek, diğer kubite işlem uygulanmayacaktır.

▪

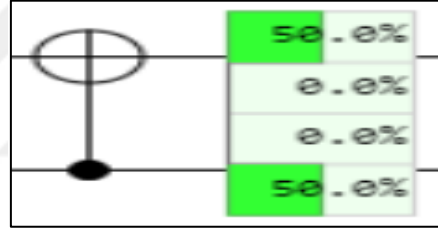


Şekil 28. İkinci Adımdaki işlemler.

Mevcut durumda girişi  $\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$  hesaplamıştık. Hadamard ve birim matrisini tensor çarpımını alarak bütünleşik operatörü hesaplayabiliriz. Elde edilen 4x4 operatör durum vektörüne uygulandığında, sonuç vektörünü aşağıdaki gibi buluruz. Bulunan vektörün ölçüm değeriyle uyduğunu gözlemlemekteyiz.

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} * \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{2}{\sqrt{2}} \\ 0 \\ \frac{2}{\sqrt{2}} \\ 0 \end{pmatrix}$$

- Üçüncü Adımda bir kontrol-not kapısı kullanılmaktadır

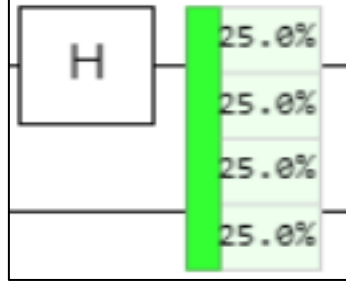


Şekil 29. Üçüncü Adım CNOT operatörü.

Mevcut durumumuz,  $\begin{pmatrix} \frac{2}{\sqrt{2}} \\ 0 \\ \frac{2}{\sqrt{2}} \\ 0 \end{pmatrix}$  CNOT operatöründe işlediğimizde durum vektörümüzü aşağıdaki değeri alacaktır.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} \\ 0 \\ 0 \\ \frac{\sqrt{2}}{2} \end{pmatrix}$$

- Dördüncü adımda yine Hadamard kapısı ilk kubiti işlemektedir. Diğer kubit işlenmeden diğer kapıya aktarılır.

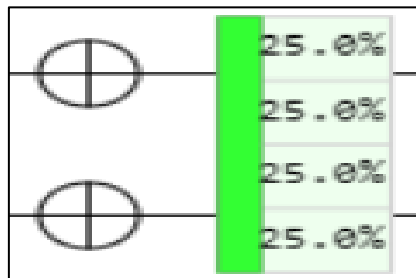


Şekil 30. Dördüncü Adım.

Bu kapıya giriş durum vektörü:  $\begin{pmatrix} \frac{\sqrt{2}}{2} \\ 0 \\ 0 \\ \frac{\sqrt{2}}{2} \end{pmatrix}$  Hadamard ve birim matrisi ile çarparak aşağıdaki sonucu elde ederiz.

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} * \begin{pmatrix} \frac{\sqrt{2}}{2} \\ 0 \\ 0 \\ \frac{\sqrt{2}}{2} \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ \frac{-\sqrt{2}}{2} \end{pmatrix}$$

- Beşinci Adım:  
Mevcut durumdaki vektörleri iki NOT kapısından geçirdiğimizde



Şekil 31. Beşinci Adım.

Mevcut durum vektörü  $\begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \end{pmatrix}$

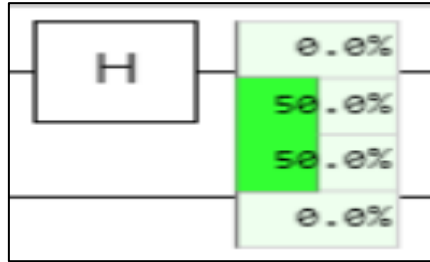
Öyle bir çıkış gibi olacak.

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \end{pmatrix} = \begin{pmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}$$

- Altıncı adım

Beşinci adımdan çıkan Hadamard ve birim matrisi ile çarparak aşağıdaki çıkışı elde ederiz.

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} * \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \\ \frac{2}{\sqrt{2}} \\ \frac{2}{\sqrt{2}} \\ 0 \end{pmatrix}$$

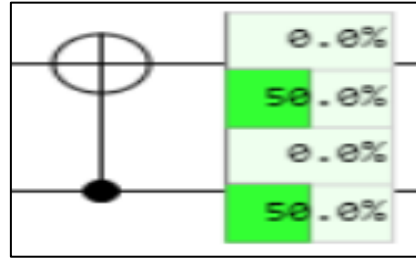


Şekil 32. Altıncı adımdan sonra durum vektörü.

- Yedinci adım:

Bu aşamada yine CNOT operatörü kullanılmaktadır.

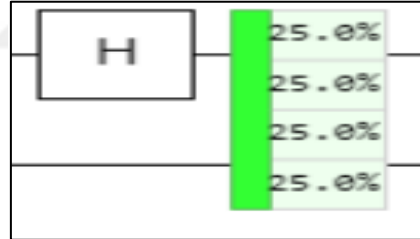
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} * \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ -2 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \\ 0 \\ \frac{\sqrt{2}}{2} \end{pmatrix}$$



Şekil 33.Yedinci adım sonrasında sistemin durumu.

- Sekizinci adım: Burada yine beşinci adımdaki işlem mevcut vektöre uygulanmaktadır.

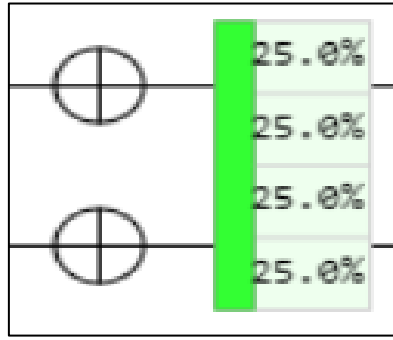
$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} * \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ -2 \\ 0 \\ 2 \end{pmatrix} = \begin{pmatrix} -\frac{2}{\sqrt{2}} \\ \frac{2}{\sqrt{2}} \\ \frac{2}{\sqrt{2}} \\ -\frac{2}{\sqrt{2}} \end{pmatrix}$$



Şekil 34.Sekizinci adım.

- Dokuzuncu adım: her iki kubitin tersinin alınması işlemi

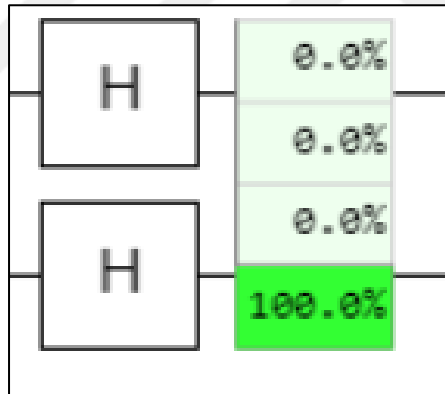
$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} * \frac{1}{\sqrt{2}} \begin{pmatrix} -2 \\ 2 \\ 2 \\ -2 \end{pmatrix} = \begin{pmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}$$



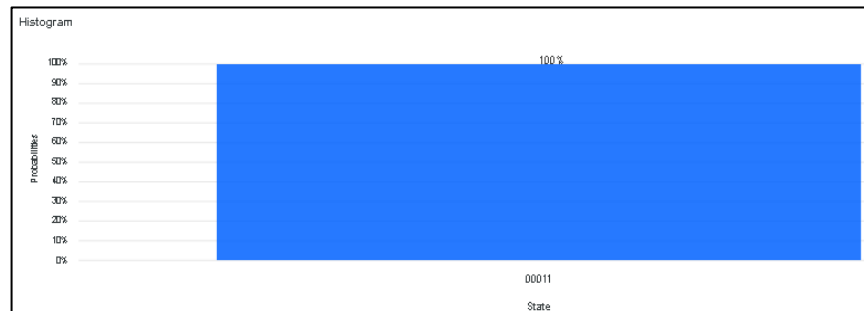
Şekil 35. Dokuzuncu adımda kubitlerin tersi alınır.

- Son işlemde her iki kubit hadamard kapılarında işlenir ve ölçüm yapılır.

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} * \frac{1}{\sqrt{2}} \begin{pmatrix} -2 \\ 2 \\ 2 \\ -2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \end{pmatrix}$$



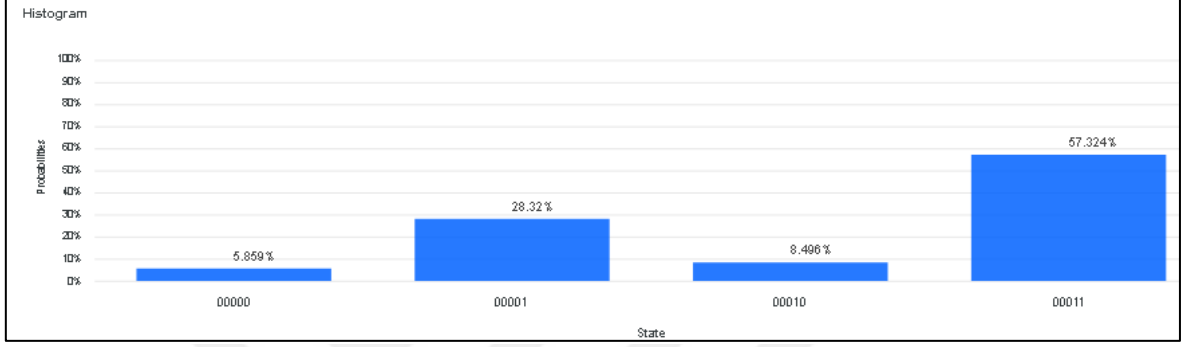
Şekil 36. Onuncu adımda Hadamard kapıları kullanılmaktadır.



Şekil 37. Grover algoritmasını simülasyonda çalıştırdığımızda.



Sadece ket 11 de ölçüldüğünü görmekteyiz. Şekil 36' da bu açıkça görülmektedir. Ancak York town daki 5 kubitlik kuantum bilgisayarıda ölçüm yaptığımızda ket 11'in ölçülme ihtimalininin yüksek olduğunu gözlemlemekteyiz. Diğer durumların da ölçüm sonucunda gözükmesi sistemin koherence (bozulma) olduğundan kubit değerlerininin uzun süre saklanamamasıdır (bkz. Şekil 37).



Şekil 38. IBM Yorktown, 5 kubitlik kuantum bilgisayarıda Grover algoritmasının çalıştırılması sonucu.

## 5. SONUÇLAR

QISKIT ile x kapısı uyguladık. verilen kubit değerini tersine çevirdi. Tasarımımızı Yorktown'daki IBM kuantum bilgisayar üzerinde test ettik .

Y kapısı Bloch Küresi üzerinde döndürme işlemi yaptı. X kapıdan farklı olarak Y eksenini üzerinde döndürme işlemi yapılır. z kapısı Bloch küresinde Z eksenini üzerinde döndürme işlemi yaptı.

kuantum bilgisayarlar çözülemeyen sorunlara özel bir çözüm bulmak için tasarlanmıştır ve karmaşık hesaplamalar hesaplayabilmek için gerçekleştirdi.

Grover Arama algoritması kuantum bilgisayar üzerinde daha hızlı çalışıyor demek kuantum bilgisayar kalasık bilgisayardan dah hızlıdır.

## 6. ÖNERİLER

1. Kalan kapıları cont, swap, ccnot, cwap, hadamard, QISKIT kullanarak, kod yazması ve değer simülatörleri QU Touch şirketinin bir ürünü olan QX gibi kullanarak kuantum kapılarını simüle edebilir.
2. 2 kubitlik kullanarak simüle ettik ama gelecekte daha fazla kubit kullanarak simüle edebilir.



## 7. KAYNAKLAR

- Akama, S., 2015. Elements of quantum computing, Springe.
- Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D. P., Margolus, N., Shor, P., Sleator, T., Smolin, J. A. ve Weinfurter, H., 1995. Elementary gates for quantum computation, Physical review A, 52,5, 3457.
- Ben-Aryeh, Y., 2009. Simulation of quantum gates by postselection of interference experiments in multi-port beam-splitter (BS) configurations, arXiv preprint arXiv:0909.3970.
- Benioff, P., 1982. Quantum-Mechanical Models of Turing-Machines That Dissipate No Energy, Physical Review Letters, 48,23, 1581-1585.
- Cross, A. W., Bishop, L. S., Smolin, J. A. ve Gambetta, J. M., 2017. Open quantum assembly language, arXiv preprint arXiv:1707.03429.
- Deutsch, D., 1985. Quantum theory, the Church–Turing principle and the universal quantum computer, Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 400,1818, 97-117.
- Deutsch, D. E., 1989. Quantum computational networks, Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 425,1868, 73-90.
- Dowd, K. ve Severance, C., 2010. High performance computing.
- Eijkhout, V., 2013. Introduction to High Performance Scientific Computing, Lulu. com.
- Feynman, R. P., 1982. Simulating Physics with Computers, International Journal of Theoretical Physics, 21,6-7, 467-488.
- Grover L.K.: A fast quantum mechanical algorithm for database search, Proceedings, 28th Annual ACM Symposium on the Theory of Computing, (May 1996) p. 212.
- Hayes, J. P., Annual ACM IEEE Design Automation Conference: Proceedings of the 40 th conference on Design automation 2003, Tutorial: basic concepts in quantum circuits, 2: 893-893.
- Khammassi, N., 2016. QX Quantum Code 0.1 User Manual, 1nd, QuTech, Computer Engineering Lab TU Delft, The Netherlands.
- Khammassi, N., Ashraf, I., Fu, X., Almudéver, C. G. ve Bertels, K., Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017 2017, QX: A high-performance quantum computer simulation platform: 464-469.

- Mlnařík, H., 2007. Quantum programming language LanQ, Masarykova univerzita, Fakulta informatiky.
- Nielsen, M. A. ve Chuang, I., Quantum computation and quantum information. (2002).
- Patvardhan, C., 2019. Quantum Computing(from the unthinkable to the inevitable), Quantum Computing(from the unthinkable to the inevitable), Dayalbagh Educational Institute.
- Phillips, A. C., 2013. Introduction to quantum mechanics, John Wiley & Sons.
- Schaller, R. R., 1997. Moore's law: past, present and future, IEEE spectrum, 34,6, 52-59.
- Shor, P. W., Proceedings 35th annual symposium on foundations of computer science 1994, Algorithms for quantum computation: Discrete logarithms and factoring: 124-134.
- URL-1, <https://www.udemy.com/qc101-introduction-to-quantum-computing-quantum-physics-for-beginners/learn/lecture/11430330#content> QC101 Quantum Computing & Quantum Physics for Beginners. 15 Temmuz, 2019.
- URL-2, <https://quantum-computing.ibm.com/support/guides/introduction-to-quantum-circuits?page=5cae6f7735dafb4c01214bbe#> Introduction to Quantum Circuits.
- URL-3, <https://algassert.com/quirk#> Welcome to Quirk. 31 15 Temmuz, 2019.
- URL-4, <http://quantum-studio.net/> Nader Khammassi The QX Simulator. 20 Temmuz, 2019.
- URL-5, <http://bilgisayarkavramlari.sadievrenseker.com/2010/10/14/kuantum-kapilari-quantum-gates/> . Bilgisayar Kavramları. 16 Temmuz, 2019.
- URL-6, <https://github.com/Strilanc/Quirk/wiki/How-to-use-Quirk> Strilanc/Quirk. 01 Ağustos, 2019.
- URL-7, <https://algassert.com/quirk#> Quirk. 01 Ağustos, 2019.
- URL-8, <https://byjus.com/physics/electromagnetic-radiation>. Byju's Learning App. 16 Temmuz, 2019.
- Williams, C. P., 2010. Explorations in quantum computing, Springer Science & Business Media.

Yao, A. C.-C., 1993 Proceedings of IEEE 34th Annual Foundations of Computer Science Quantum circuit complexity: 352-361.



## ÖZGEÇMİŞ

Fathelrhman Mohammed ABKER MOHAMMED, 1990 Wad Madani, Sudan doğumludur. İlk ve orta öğretim eğitimini Alzaiem Alazhari Okulu'nda ve lise eğitimini "Mubarak Zarrouk, Fateh, Wadmadani Al Amireya" Okullarında tamamlamıştır. 2010 yılında Kuran-ı Kerim ve İslami Bilgiler Üniversitesi Bilgi sistemleri Bölümü'nde başladığı lisans eğitiminden 2014 yılında mezun olmuştur. Eğitime devam edebilmek için 2015 yılında Türk Diyanet İşleri Başkanlığı tarafından Kur'an-ı Kerim derslerinin bursunu kazanmış olup, 2016 yılında hafızlık tespit sınavına girerek tamamlamıştır. Aynı yılın Eylül ayında " Türkiye bursları" tarafından burs kazandı. 2016 - 2017 Eğitim öğretim yılında Karadeniz Teknik Üniversitesi Türkçe Öğretim Uygulama ve Araştırma Merkezi'nde Türkçe hazırlık öğrenimi gördü. 2017 Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü; Bilgisayar Mühendisliği Bölümü'nde yüksek lisans eğitimine başlamıştır.

ABKER MOHAMMED, kendi ana dili dahil 4 dil bilmektedir. Pular (anadil), Arapça, İngilizce (Orta seviye) ve Türkçe.

- 13 Ağu 2013–27 Ağu 2013 Su Kaynakları ve Sulama ve Elektrik Bakanlığı, Khartoum, Sudan bilgi teknisyeni olarak çalışmıştır.
- 25 Kasım 2014–24 Mayıs 2015 Sertifika Ofisi Koordinatör Yardımcısı Kur'an-ı Kerim ve bilimin pekiştirilmesi Üniversitesi, Wad Madani (Sudan).