



**T.C.  
İSTANBUL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**



**YÜKSEK LİSANS TEZİ**

**W-STATE KUANTUM AĞLARININ GELİŞTİRİLMESİ**

**Sinan BUĞU**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Bilgisayar Mühendisliği Programı**

**Danışman**

**Prof. Dr. Ahmet SERTBAŞ**

**II. Danışman**

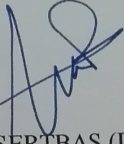
**Doç. Dr. Fatih ÖZAYDIN**

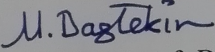
**Haziran,2015**

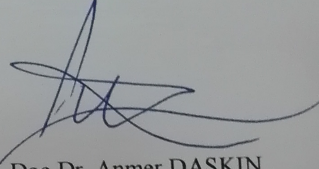
**İSTANBUL**

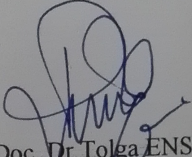
Bu çalışma 15/06/2015 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği programında Yüksek Lisans Tezi olarak kabul edilmiştir.

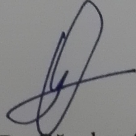
**Tez Jürisi:**

  
Prof. Dr. Ahmet SERTBAŞ (Danışman)  
İstanbul Üniversitesi

  
Yrd. Doç. Dr. Mustafa DAĞTEKİN  
İstanbul Üniversitesi

  
Yrd. Doç. Dr. Anmer DAŞKIN  
İstanbul Medeniyet Üniversitesi

  
Yrd. Doç. Dr. Tolga ENSARI  
İstanbul Üniversitesi

  
Yrd. Doç. Dr. Oğuzhan ÖZTAŞ  
İstanbul Üniversitesi

## ÖNSÖZ

Kişisel ve akademik gelişimime yapmış olduğu katkılarından dolayı Prof. Dr. Sıddık YARMAN'a ve destekleri için FMV Işık Üniversitesi BAP-14A101 projesine teşekkür ederim.

Haziran,2015

Sinan BUĞU

# İÇİNDEKİLER

Sayfa No

ÖNSÖZ.....	i
İÇİNDEKİLER .....	ii
ŞEKİL LİSTESİ.....	iii
ÖZET.....	v
SUMMARY .....	vi
1. GİRİŞ.....	1
2. GENEL KISIMLAR .....	7
3. MALZEME VE YÖNTEM .....	11
3.1. PARAMETRELER .....	11
3.1.1. Kaynak Maliyeti .....	11
3.1.2. Birleştirilen Ağ Sayısı .....	12
3.2. YÖNTEM .....	13
3.3. SİMÜLASYONDA KULLANILAN PROGRAMLAR VE DİLLER .....	17
4. BULGULAR .....	22
5. TARTIŞMA VE SONUÇ .....	27
KAYNAKLAR .....	29
EKLER.....	33
EK 1. Simülasyonlarda ve grafik çizimlerinde kullanılan programlama, script dilleri ve yardımcı araçlara ait bazı kod ve script örnekleri .....	33
ÖZGEÇMİŞ.....	49

## ŞEKİL LİSTESİ

### Sayfa No

Şekil 1.1: 1989 yılında IBM'in 35 xenon atomuyla yazdığı ismi. ....	2
Şekil 1.2: DWave şirketinin ticari ürün olarak sattığı, Google ve NASA ortaklığındaki Kuantum Yapay zeka Laboratuvarı'nda kullanılan süperiletken teknolojiyle Geliştirilmiş 521 kuantum tranzistörlü kuantum bilgisayar. ....	3
Şekil 1.3: IBM'in 29 Nisan 2015'te tanıttığı yeni kuantum çipi. Mili-Kelvin sıcaklığına ulaşıldığında, devrenin ortasındaki dört koyu bölge, kuantum mekaniksel olarak digital bilgiyi işleyebilmektedir. ....	4
Şekil 2.1: Tashima ve diğ. (2009). ....	8
Şekil 2.2: Tashima ve diğ. (2009). ....	8
Şekil 2.3: Tashima ve diğ. (2009). ....	9
Şekil 2.4: Tashima ve diğ. (2009). ....	9
Şekil 2.5: Tashima ve diğ. (2009). ....	9
Şekil 2.6: Ozdemir ve diğ. (2011). ....	10
Şekil 2.7: Ozdemir ve diğ. (2011). ....	10
Şekil 3.1: Ozdemir ve diğ. (2011). ....	12
Şekil 3.2: Deneysel düzeneklerdeki devre çizimlerinde Pauli- X kapısı HWP , Hadamard kapısı QWP vs. Olarak gösterilmektedir. ....	13
Şekil 3.3: Dört kuantum bitlik bir devrede, CNOT ve NOT (Pauli-X) kapılarının alternatif kullanım sonuçlarımız. ....	14
Şekil 3.4: İki'den fazla W ağını birleştirebilmeyi denemek için hazırladığımız bir simülasyonun ekran çıktısı. ....	15
Şekil 3.5: Fotonik W ağlarını birleştirmek üzere geliştirdiğimiz bir kuantum optik devresi. ....	16
Şekil 3.6: Mevcut devre (kırmızı eğri, Ozdemir ve diğ. 2011) ile bir deneme devremizin (mavi eğri) kaynak maliyet performansı açısından karşılaştırılması. ....	16
Şekil 3.7: C dilinde yazılmış simülasyon çıktısı. ....	18
Şekil 3.8: C++ dilinde yazılmış simülasyon. ....	19
Şekil 3.9: Python'da yazılmış simülasyon. ....	20
Şekil 3.10: Mathematica'da yazılmış simülasyon ve çizdirilen grafikler. ....	21
Şekil 4.1: Fredkin gate kullanarak tasarladığımız devre . Bugu ve diğ.(2013). ....	22
Şekil 4.2: Yeni devremizin başarı olasılık tablosu Bugu ve diğ.(2013). ....	22

<b>Şekil 4.3:</b> Yeni devremizin 3 qubitlik 2 W state'i (Bell state'leri) ancilla foton kullanarak (sarı renkli) birleştirmesini gösteriyor. Bugu ve diğ.(2013). .....	23
<b>Şekil 4.4:</b> Toffoli gate kullanarak geliştirdiğimiz devre. Yeşilyurt ve diğ. (2013). .....	23
<b>Şekil 4.5:</b> Geliştirdiğimiz devrenin W state'leri birleştirme başarı olasılık tablosu. Yeşilyurt ve diğ. (2013). .....	24
<b>Şekil 4.6:</b> Ancilla foton ve Fredkin gate kullanarak geliştirdiğimiz Fusion(birleştirme) devresi. ....	25
<b>Şekil 4.7:</b> Fredkin gate ve ancilla foton kullanarak geliştirdiğimiz devrenin tek seferde 3 adet W4 kullanarak W9 elde etmesini gösteren şekil. Ozaydin ve diğ. (2014). .....	25
<b>Şekil 4.8:</b> Geliştirmiş olduğumuz devrenin başarı olasılık tablosu. Ozaydin ve diğ. (2014). .....	26
<b>Şekil 4.9:</b> Grafikte, Kırmızı renk (Can ve diğ.2013) in, mavi renk son olarak sunmuş olduğumuz devrenin,siyah renk Özdemir ve diğ. 2011 in, siyah renk aynı zamanda ilk bulduğumuz olan Bugu ve diğ. 2013'ün maliyetini gösteriyor. Grafik için 1000 adet birleştirme denemesi denemesi yapıp ortalamaları alınmıştır . Ozaydin ve diğ. (2014). .....	26

## ÖZET

### YÜKSEK LİSANS TEZİ

### W-STATE KUANTUM AĞLARININ GELİŞTİRİLMESİ

Sinan BUĞU

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman : Prof. Dr. Ahmet SERTBAŞ

II. Danışman : Doç. Dr. Fatih ÖZAYDIN

İki tarafın paylaştığı iki parçacık arasındaki “klasik ötesi” ilişki, yani kuantum dolanıklığın (bi-partite entanglement) fiziksel üretimi, bilgi kuramsal içeriği ve gerek haberleşme mühendisliğinde gerek bilgisayar hesaplamalarında kullanılması konuları çokça çalışılmıştır. Ancak çok taraflı dolanıklık (multi-partite entanglement) henüz iki taraflı dolanıklık kadar keşfedilememiştir. Bilgi işleme, hesap yapma, ışınlama, haberleşme ve kriptografi gibi birçok konunun her birinde, çok taraflı dolanık ağların başlıcaları olan GHZ, Cluster ve W tarzı sistemlerin bazıları özellikle gerekmektedir. Çok taraflı kuantum yapılar üretilip taraflar arasında başarıyla paylaştırıldığında, aslında, mesela GHZ tarzı ya da W tarzı bir “kuantum network(kuantum ağ)” inşa edilmiş olmaktadır. Dolayısıyla her birçok taraflı kuantum dolanık bir yapı (mesela bir W state), bir kuantum ağ olarak görülebilir. İstenen boyutta GHZ ve Cluster ağlarının oluşturulması problemi çözülmüş olsa da, W ağı için bu problem henüz çözülememiştir. Bu yüzden çok taraflı W state tarzı kuantum ağların düşük maliyetle ve yüksek başarı oranıyla elde edilebilmesi, kuantum çağı için büyük önem arz etmektedir. Bu tez kapsamında, yüksek boyutlu W çalışmada yazılım ve simülasyon gibi bilgisayar mühendisliğinin alt kollarından faydalanılarak yüksek verimli kuantum ağlarının nasıl elde edildiği gösterilmiştir.

Mayıs 2015,56 Sayfa.

**Anahtar kelimeler:** Kuantum Ağlar, W- State, Dolanıklık.

## **SUMMARY**

**M.Sc. THESIS**

**ENHANCING W-STATE QUANTUM NETWORKS**

**Sinan BUĞU**

**İstanbul University**

**Institute of Graduate Studies in Science and Engineering**

**Department of Computer Engineering**

**Supervisor : Prof. Dr. Ahmet SERTBAŞ**

**Co-Supervisor : Asst. Prof. Dr. Fatih ÖZAYDIN**

The production and the manipulation, as well as the information theoretical aspects of bipartite quantum entanglement (i.e. EPR/Bell pairs) have been studied well but that is not the case for multipartite entanglement. In several quantum information tasks, specific multipartite states are required. Creating and sharing multipartite quantum networks lead to construct W or GHZ states. So, we can regard all multipartite quantum networks as quantum network. The efficient or deterministic creation of GHZ and cluster states has been studied but there is still a lot research to perform, for the W state because of its specific structure. As the size of the target W state increases, the success probability of production was decreasing exponentially therefore this was a big obstacle for performing the tasks that require W states. In this thesis our main goal is to enhance the large W state preparation process in terms of the cost via computer engineering technics and simulations.

May 2015,56 Pages.

**Keywords:** Quantum Networks, W-State, Entanglement.

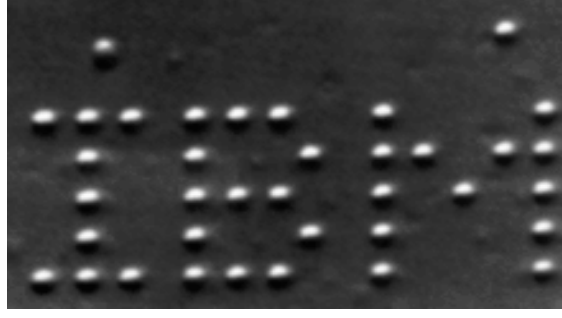


## 1. GİRİŞ

Klasik mekanik çok başarılı olmasına karşın, 1800'lü yılların sonlarına doğru, siyah cisim ışıması, tayf çizgileri, fotoelektrik etki gibi bir takım olayları açıklama da yetersiz kalmıştır. Açıklamaların yanlışlığı bilim adamlarının yetersizliğinden değil aksine klasik mekaniğin yetersizliğinden kaynaklanıyordu. Klasik mekanikteki sorun:klasik mekanik Evren'i sürekli olarak modelliyordu. Bu modelleme yanlıştı çünkü üç konum ve üç momentumla tanımlanan parçacıklar, sonsuz sayıda parametreyle tanımlanan alanlarla bir aradaydılar. Eş dağılım kuramınca ("equipartition theorem") sistemin enerjisinin denge durumunda sistem bileşenlerine eş biçimde dağılması gerekir. Alanlar sonsuz bileşene sahip olduğundan bütün enerji alanlara kalır. Bu ise eş dağılım kuramına ters düşüyordu.

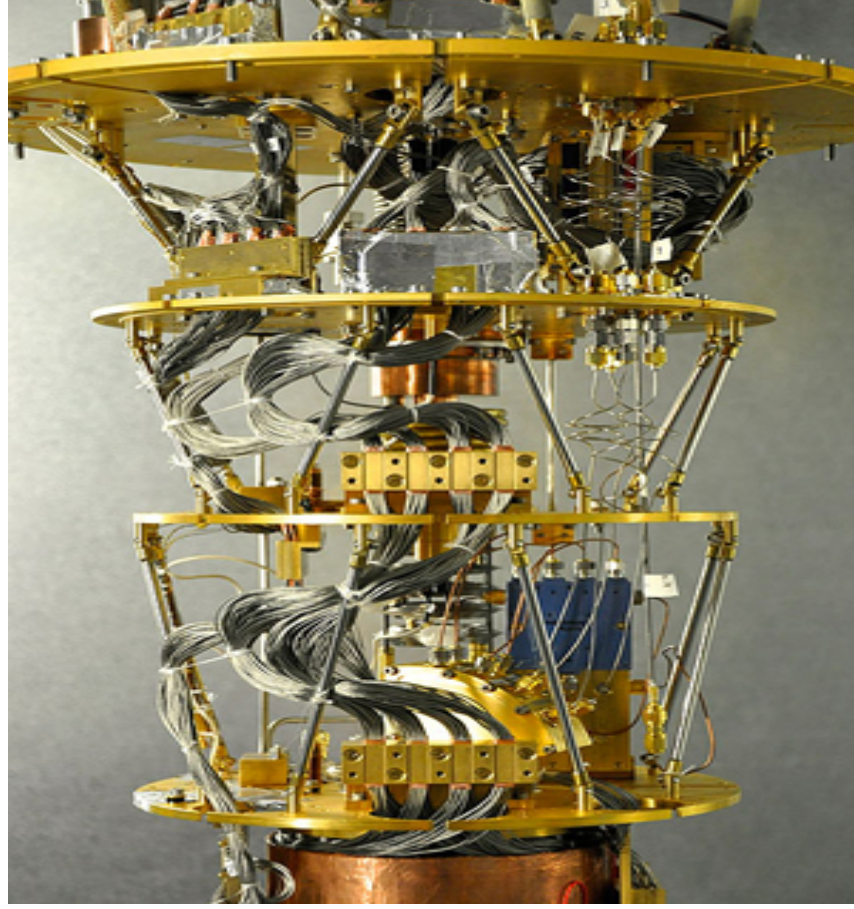
Kuantum kuramı ise olayı bambaşka bir şekilde ele alır. Parçacıklar artık doğrudan 3 konum ve 3 momentumla tanımlanmak yerine bir "dalga fonksiyonu" ile tanımlanırlar. Bu dalga fonksiyonu parçacığın bütün bilgisini içinde barındırır ve dalga fonksiyonuna uygun "sorular" sorularak gerekli bilgi alınır. Örneğin konum bilgisi için dalga fonksiyonuna "parçacık nerede?" sorusunu sorarsınız, o ise size parçacığın soruyu sorduğunuz anda nerede olabileceğini söyler. Buradaki kritik nokta olabirliktir. Bu, dalga fonksiyonunun bir de "olasılık fonksiyonu" olarak anılmasına neden olmaktadır. Bu nedenlerden dolayı klasik mekanikle açıklanamayan şeyler kuantum mekaniğiyle açıklanabilir oldu.

Kuantum bilgisayarının başlangıcı,1959 yılındaki ünlü konuşmasında Richard Feynman, "aşağıda daha çok yer var" (There's Plenty of Room at the Bottom) diyerek, sentezlenmiş kimyasal yapılar kullanmak yerine atomları tek tek kontrol etme fikriyle bir koldan genel olarak nanoteknoloji, bir koldan da kuantum bilgisayarı fikrini ortaya atmasıyla başladı. Feynman'ın bu fikrinin hayata geçirilme çabaları, ancak 1990'larda ciddi bir seviyeye gelebildi.



**Şekil 1.1:** 1989 yılında IBM'in 35 xenon atomuyla yazdığı ismi.

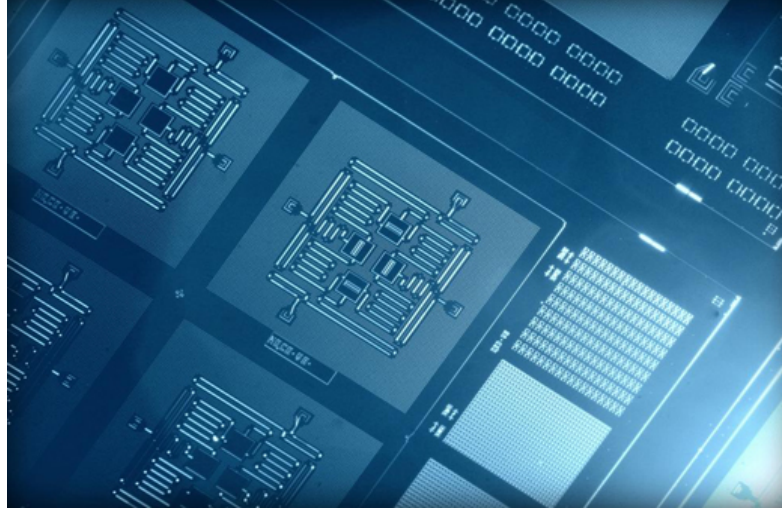
1982 yılındaki, “bilgisayarlarla fiziği simüle etmek” başlıklı kilometre taşı makalesinde, yalnızca kuantum bilgisayarı fikrini olgunlaştırmış olarak açıklamakla kalmıyor, aynı zamanda klasik bilgisayarların doğayı ve kuantum bilgisayarlarını simüle edemeyeceği, ancak kuantum bilgisayarlarının hem doğayı, hem klasik bilgisayarları simüle edebileceğini [7] anlatıyordu. 1985’de Deutsch, ilk “evrensel kuantum bilgisayarı”nı önerdi [6]. 1991’de Ekert, kuantum dolanıklık sayesinde güvenli haberleşme yapılabileceğini buldu [8]. 2000 yılında ilk kuantum bilgisayarı, NMR teknolojisi ile gerçekleştirildi [42] 2001’de ise IBM Almaden Araştırma Merkezi’nde, 15 sayısı ilk kez bir kuantum bilgisayarı ile çarpanlarına ayrıldı. Aradan geçen yaklaşık 15 sene içinde gelişen kuantum bilgisayarları, D-Wave şirketi tarafından ticari ürün haline getirildi. Bu kuantum bilgisayarı, Google ve NASA işbirliği tarafından satın alınarak, Kuantum Yapay Zeka Laboratuvarı (QuAIL) kuruldu [43].



**Şekil 1.2:** DWave şirketinin ticari ürün olarak sattığı, Google ve NASA ortaklığındaki Kuantum Yapay Zeka Laboratuvarı'nda kullanılan, süperiletken teknolojiyle geliştirilmiş 512 kuantum tranzistörlü kuantum bilgisayarı.

IBM ise 2015 İlkbaharı'nda, geliştirdiği süperiletken kuantum bilgisayarı tanıttı [44].

Microsoft ise, "Microsoft Research Station Q" merkezinde topolojik kuantum hesaplama üzerine ve "Quantum Architectures and Computation Group (QuArC)" merkezlerinde kuantum bilgisayarı mimarisi ve hesaplama üzerine çalışmalarını her geçen gün yoğunlaştırmaktadır.



**Şekil 1.3:** IBM'in 29 Nisan 2015'te tanıttığı yeni kuantum çipi. Mili- Kelvin sıcaklığına ulaşıldığında, devrenin ortasındaki dört koyu bölge, kuantum mekaniksel olarak dijital bilgiyi işleyebilmektedir.

Özetle, günümüzde, atomaltı parçacıkların (elektron, foton vs.) kuantum mekaniksel özelliklerini kullanmak, bilgisayar ve haberleşme mühendisliklerinde çığır açmaktadır. Klasik mekanikte bulunmayan başta kuantum dolanıklık (entanglement) üstkonumluluk (superposition) ve kuantum belirsizlik gibi özellikler sayesinde, klasik sistemlerle yapılması imkansız birçok işin yapılması mümkün olmaktadır. Örneğin kuantum teleportasyon, kuantum kriptografi, vs. Öte yandan, klasik sistemlerde hal-i hazırda gerçekleştirilebilen birçok algoritmanın da, kuantum sistemlerde çok daha hızlı gerçekleştirilebileceği kanıtlanmıştır. Örnek: Shor'un çarpanlarına ayırma algoritması, kuantum Fourier dönüşümü, Grover'in kuantum arama algoritması vs. Klasik kriptografi protokollerinin özü, çarpanlara ayırma gibi problemlerin (mevcut klasik fiziksel bilgisayarlarda çözülmesinin) "zorluğuna" dayanmaktadır. Ne var ki bu problemlerin, kuantum bilgisayarları için "kolay" çözülen sınıfta olduğu ortaya çıktığından, kuantum teknolojilerinin gelişmesiyle birlikte, klasik kriptografinin güvenliği ortadan kalkmaktadır. Yani kuantum teknolojilerine sahip olanlar, klasik bilgisayarlarla hazırlanan tüm kriptoları çok kolay bir şekilde kırabileceklerdir. Buna çözüm olarak, kuantum kriptografi, kuantum haberleşme ve bu işleri yapabilecek kuantum haberleşme ağlarına ihtiyaç duyulmaktadır. Kuantum haberleşme ağları, çok

tarafli dolanik (multipartite entangled) sistemler ile gercekleştirilmektedir. İki tarafın paylařmıř olduđu atomaltı iki parçacık arasındaki kuantum dolanıklık (yani iki tarafli, “bipartite entanglement”, örnek: EPR/Bell çifti), fiziksel üretim, bilgi kuramsal içerik, bilgi güvenliđi, haberleřme mühendisliđi ve bilgisayar algoritmaları açasından çokça çalışılmıřtır. Ancak, çok tarafli dolanıklık (multipartite entanglement) yani bir nevi kuantum haberleřme ađları, henüz iki tarafli dolanıklık kadar keřfedilememiřtir. Çok tarafli dolanıklık, GHZ, W, Cluster gibi çeřitli temel yapılarla olabilmektedir. Bu yapılar, klasik haberleřme ve yerel operasyonlarla (local operations & classical communications, LOCC) birbirine dönüřtürülemezdir. Bilgi işleme, hesap yapma, haberleřme ve kriptografi gibi birçok meselenin her birinde, bu üç temel dolanıklığın bazıları özellikle gerekmektedir. Örneđin:

- Cluster State’ler, ölçüm tabanlı kuantum hesaplama için evrensel tek araçtır. GHZ State’ler; kuantum ışınlamada, kuantum kriptografide, kuantum sır saklamada ve dađıtık ađlardaki birçok işte gerekmektedir ve/veya kullanılabilir.
- W State’ler ise, birçok kuantum kriptografi protokolünde ve kuantum algoritmasında verimli bir şekilde kullanılabilir gibi, optimal evrensel kuantum klonlama makinasında ve özellikle anonim kuantum ađlarında liderlik seçiminde ise bizzat gerekmektedir.

Çok tarafli kuantum yapılar üretilip, taraflar arasında başarıyla paylařıldığında, aslında, mesela GHZ ya da W sınıfından bir kuantum haberleřme ađı inşa edilmiř olmaktadır. Dolayısıyla her bir çok tarafli kuantum dolanık bir yapı (mesela W state), bir kuantum ađı olarak görülebilir. Kuantum haberleřme, kriptografi ve teleportasyonda kullanılan temel kuantum yapı foton olduđundan, bu çok tarafli yapıların fotonlarla üretilmesi ve kuantum optik elemanlarla işlenmesi gerekmektedir. Bilgi, fotonun polarizasyonunda kodlanabilmektedir; mesela fotonun kuantum durumu “yatay (H)” ve “dikey (V)” olarak kodlanabilmekte ve bilgi bu şekilde işlenebilmektedir. Çok tarafli fotonik GHZ ve

Cluster State'lerinin üretimi, nispeten çalışılmış, anlaşılmıştır ve deterministik üretim süreçleri geliştirilmiştir. Ne var ki, özel yapısından dolayı çok taraflı fotonik W State'lerinin üretimi yeni yeni keşfedilmektedir. W state'in taraf sayısı (state'in boyutu) büyüdükçe, bilinen yöntemlerle üretimin başarıya ulaşma ihtimali üssel olarak düşmekteydi. Bu da W state'in gerektiği kuantum algoritmalar ve diğer kuantum işlerin gerçekleştirilebilmesi önünde büyük bir engeldi. Son yıllarda Şahin Kaya Özdemir ve grubu (M. Koashi, N. Imoto ve öğrencileri), bu konuda birçok çalışma yapmıştır ve ağdaki taraf sayısı artarken başarı ihtimalinin üsselden daha az düşebileceği bir takım yöntem ve stratejileri kuramsal olarak geliştirmiş ve bazılarını deneysel olarak da gerçekleştirmiştir[32,33,34,35]. Bu yöntem ve stratejileri, sürecin başarı ihtimaliyle ilişkilendirerek "maliyet" tanımlanmıştır. W ağlarının gerçekte oluşturulabileceğinin aslında mümkün olduğu gösterilmiş olmasına ve bu konuda bir çığır açılmış olmasına rağmen, bu yöntemlerin hiçbiri optimum değildir ve maliyetleri oldukça yüksektir. Dolayısıyla yeni yöntemler geliştirilmesi gerekmektedir.

Bu çalışmada W kuantum haberleşme ağlarının üretilmesine yönelik yeni yöntemler geliştirdik. Temel bir yaklaşımımız, Ş. K. Özdemir ve grubunun basit kuantum devresine ek olarak; başta CNOT, Hadamard, Toffoli ve Fredkin gibi ana kuantum mantıksal kapıları ve bu kapıların, modelleyeceğimiz kuantum devrelerde sağlayabileceği muhtemel yararları araştırmak ve incelemektir. Diğer bir yaklaşımımız kendi modellerimizle maliyet verimli olarak hazırlayacağımız birçok ufak W ağını tek seferde, yüksek başarı ihtimaliyle birleştirecek kuantum devreler kurmak. Kuracağımız modellerin devre maliyetlerini Wolfram Mathematica'da hazırlayacağımız simülasyonlarla, devrelerin başarı ihtimalini ve oluşturulacak W ağlarının kaynak maliyetlerini Monte Carlo simülasyonlarıyla ortaya çıkarttık.

## 2. GENEL KISIMLAR

Başta Einstein ve diğ.[7] ve Bell [2,3] tarafından, çokça tartışılan ve literatürde EPR/Bell çifti adıyla anılan iki taraflı kuantum dolanık yapının (bipartite quantum entangled) çok taraflı uzantısı (multipartite extension) olarak geliştirilen GHZ ve W durumları (diğer bir deyişle, kuantum ağları) hesaplama güçleri açısından çalışılmış [9] ve gösterilmiştir ki bu kuantum kaynaklar sayesinde birçok hesaplama problemi, klasik kaynaklara nazaran çok daha hızlı çözülebilmeyen ötesinde, birçok problem de yalnızca bu kuantum kaynakların varlığıyla çözülebilmektedir. Örneğin, anonim kuantum ağlarındaki liderlik seçimi problemi, ancak ve ancak, istenen boyutta bir W kuantum ağının varlığı ile çözülebilmektedir. Dolayısıyla GHZ ve W gibi kuantum kaynakların hazırlanması, kuantum çağında önemli bir mesele haline gelmiştir. Kullanılan teknolojiye bağlı olarak, bu kaynaklar (kuantum ağlar) çeşitli şekillerde üretilmeye çalışılmaktadır. W ağların hazırlanışı tuzaklanmış iyonlarla [16,18,30], NMR ile [36] ve en çok da optik sistemlerde fotonlarla üretilmeye ve işlenmeye çalışılmıştır [17,19,20,29,31,37,41].

W State, 4 adet Bell çiftinden birinin özel halidir. Kuantumda kullanılan aynı anda lojik olarak hem 1 hem 0 hem de her iki hali de aynı anda sağlayabilen yani süperpoze durumunda olan (toplamda 3 halde de bulunabilen) bit'lere quantum bit manasına gelen qubit denir. Fotonların yatay (Horizontal) ve dikey (Vertical) polarizasyonlarına lojik olarak 0 ve 1 dersek, 2 qubit'den oluşan W-state

$$W_2 = (|10\rangle + |01\rangle) / \sqrt{2} \quad \text{veya} \quad W_2 = (|VH\rangle + |HV\rangle) / \sqrt{2}$$

şeklinde gösterebilir.

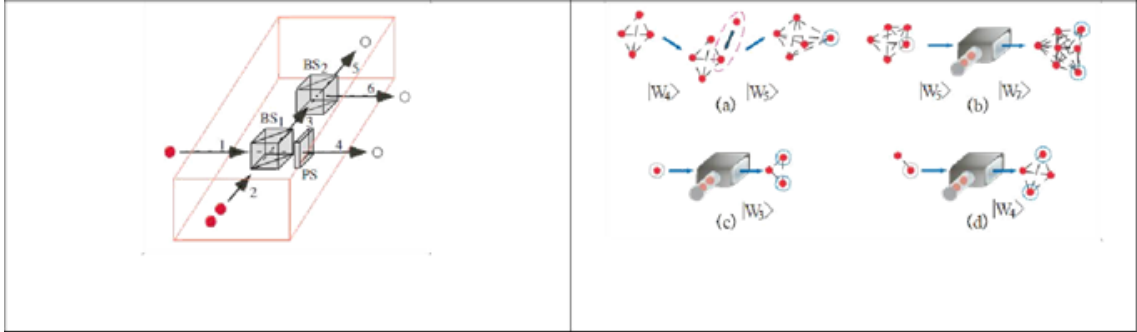
Genel olarak W state'de yalnızca 1 tane 1 (başka bir gösterimle V) bulunur ve geri kalanlar ise 0 (H) polarize fotonlardır fakat bizler hangisinin 1 hangisinin 0 olduğunu bilmeyiz. Öğrenebilmek için ölçüm yapmamız gerekir ki bu da kuantumluk durumunun çökmesine neden olur. Kuantum optik'te fotonlar kullanıldığı için burda fotonların polarizasyonundan bahsediyoruz. Oysa ki elektronlarla çalışıyor olsaydık bu defa

eletronun iç açısal momentumunun yönüne göre spin up (lojik olarak 1'e tekabül eder) ve spin down (0) terimlerinden bahsediyor olurduk. W state'in içerdiği superpoze durumundaki foton sayısına göre genelleştirilmiş formülü,

$$W_n = (|100\dots0\rangle + |01000\dots0\rangle + \dots + |00\dots01\rangle) / \sqrt{n}$$

şeklinde ifade edilir.

W ağları karmaşık yapısından dolayı ancak ufak boyutta hazırlanabilmekteyken, son yıllarda, büyük boyutta hazırlama çabaları hız kazanmıştır. W ağlarının hazırlanışındaki temel problem, hedef boyut büyüdükçe, hazırlama sürecinin başarı ihtimalinin üssel olarak düşmesiydi. Bu şartlar altında, W ağlarının hazırlanmasındaki kilometre taşı sayılabilecek çalışmaları, Osaka Grubu'ndan Ş.K. Özdemir, T. Yamamoto, M. Koashi, N. Imoto ve öğrencileri T. Tashima gerçekleştirmiştir. Özetle, Tashima ve diğ. [33] önerdikleri optik devre ile (Şekil.2.1), mevcut bir W ağının olasılıksal olarak, Şekil 2.2'deki yöntemle birer foton ile genişletilebileceğini göstermişlerdir.

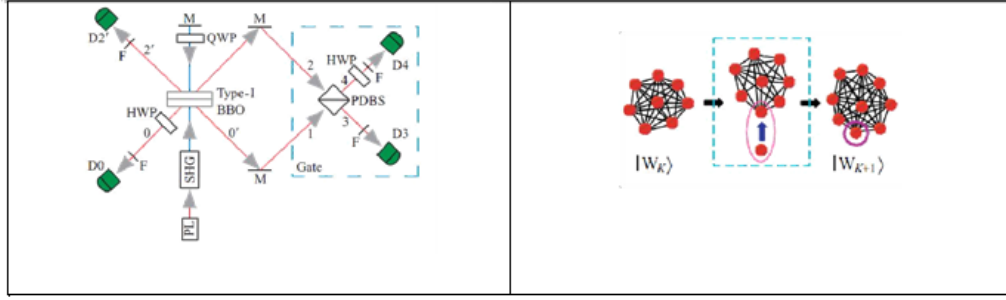


Şekil 2.1: Tashima ve diğ. (2009)

Şekil 2.2: Tashima ve diğ. (2009)

Tashima ve diğ. [34] daha sonra, yine güncel fotonik teknolojiyle rahatlıkla gerçekleştirilebilecek yeni bir devre geliştirerek (Şekil 2.3) W ağlarının genişletilmesinde (Şekil 2.4) yeni bir adım atmışlardır. Tashima ve diğ. [32] deneysel olarak, optik bir devreyle (Şekil 5), mevcut bir W ağının ikişer fotonla genişletilmesini (Şekil 2.4) gerçekleştirmiştir

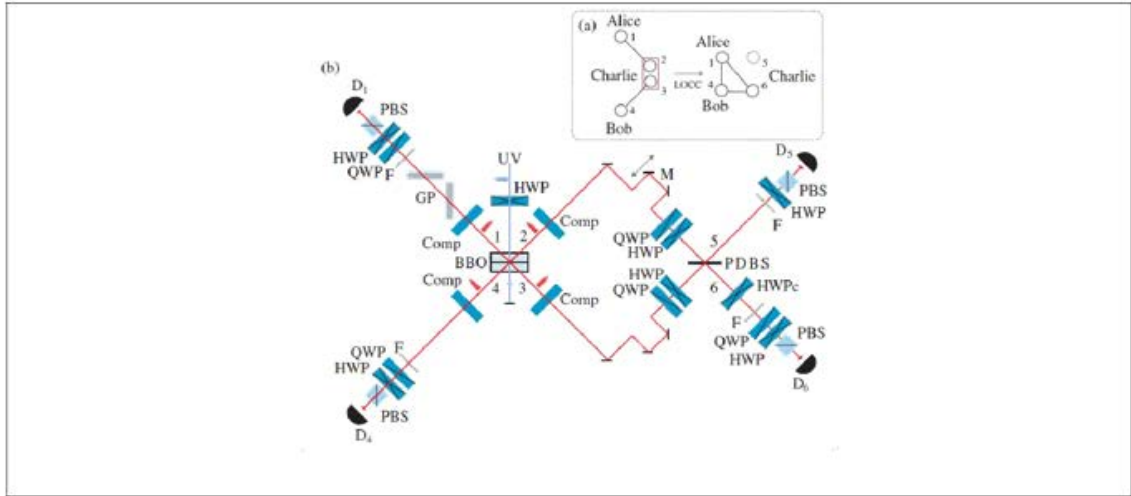




Şekil 2.3: Tashima ve diğ.(2009)

Şekil 2.4: Tashima ve diğ.(2009)

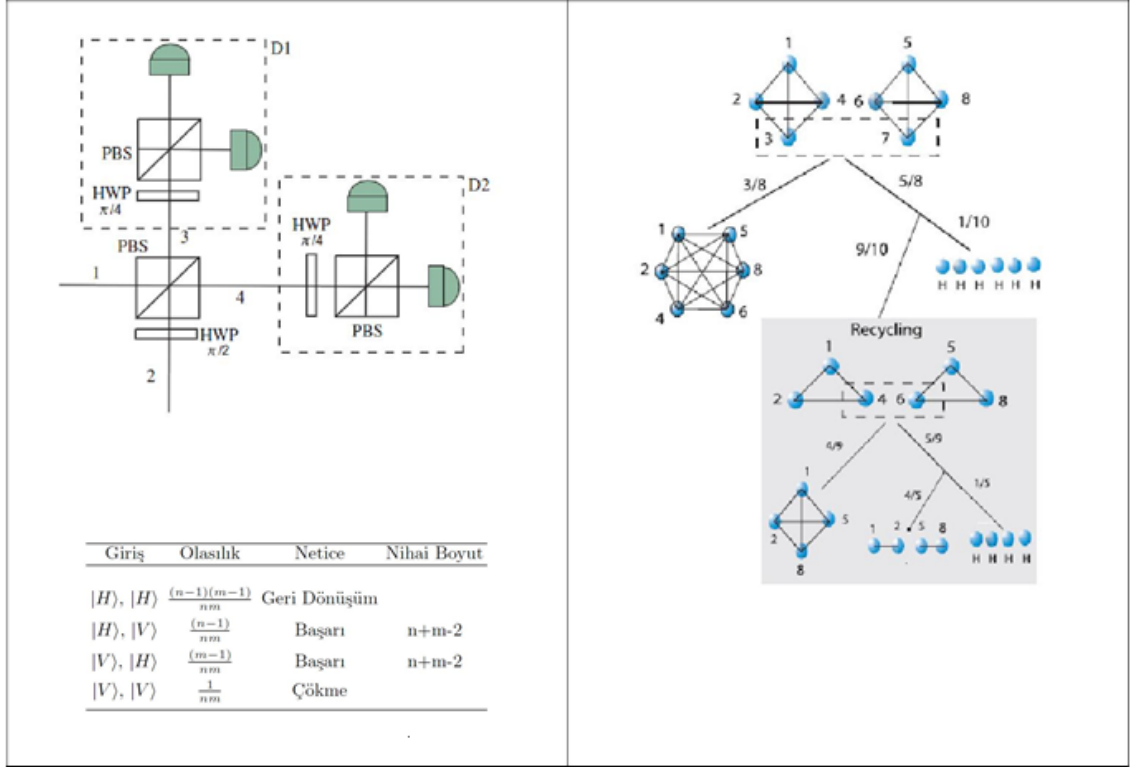
Tashima ve diğ. [35] öte yandan, iki adet Bell çiftini birleştirerek, bir adet 3 fotonlu W ağı oluşturmayı (Şekil 2.4) deneysel olarak gerçekleştiren bir devre de (Şekil 2.5) geliştirmiştir.



Şekil 2.5: Tahsima ve diğ. (2009)

Ozdemir ve diğ. [28] W ağları oluşturmadaki en önemli adımı, hedef boyut arttıkça kaynak maliyetinin üsselden daha az arttığı bir birleştirme devresi (Şekil 2.6, Şekil 2.7) olarak atmıştır. Herhangi boyuttaki iki adet W ağından birer foton, devreye 1. ve 2. kanallardan verildiğinde, iki dedektörün aynı anda yanması durumunda, iki W ağı birleşmiş ve daha büyük bir W ağı oluşturmuş olmaktadır. Dedektörlerden yalnızca birisi yandığında W ağları birleşmemiş ve kaybedilmiş (kuantum mekaniksel çökme) olmakta, yalnızca diğeri yandığında ağlar yine birleşmemiş ancak kaybedilmemiş

olarak, geri dönüşüme yapılabilecek durumda kalmaktadırlar. Başarılı birleşme durumunda dahi, iki adet dedektörde kaybolan fotonlardan dolayı,  $m$  ve  $n$  boyutunda iki adet ağ birleşince, oluşan ağın boyutu  $n+m-2$  olmaktadır. Doğruluk tablosunu Şekil 2.6'da verdiğimiz işte bu mevcut en basit ve en iyi kuantum devre bile, başarısız birleştirme ihtimallerinden dolayı, istenen verimde değildir.



Şekil 2.6: Ozdemir ve diğ. (2011)

Şekil 2.7: Ozdemir ve diğ. (2011)

### 3. MALZEME VE YÖNTEM

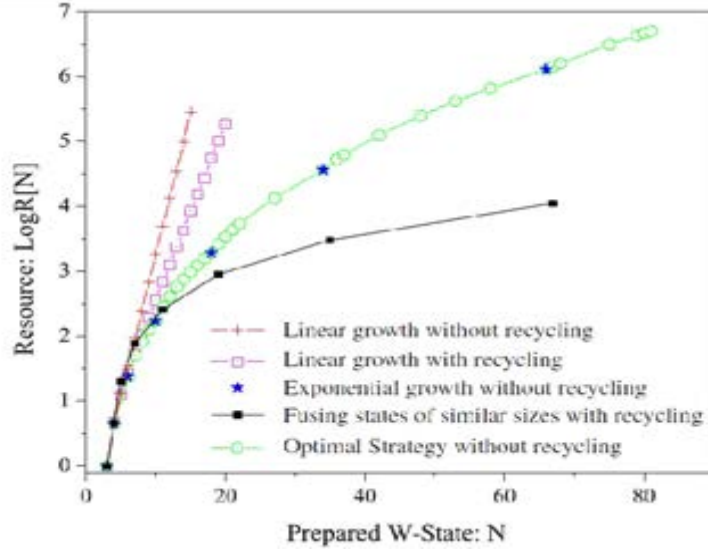
#### 3.1. PARAMETRELER

##### 3.1.1- Kaynak maliyeti

Araştırmamızda kullanacağımız ilk parametre, “kaynak maliyetidir” (resource cost).  $n$  ve  $m$  boyutundaki iki adet  $W$  ağını (yani  $W_n$  ve  $W_m$ ) birleştirerek elde edilecek  $W_{m+n-2}$  ağının maliyeti, harcanan  $W_n$  ve  $W_m$  ağlarının toplam maliyetinin ( $R[W_n] + R[W_m]$ ), başarılı birleştirme olasılığına  $P_s(W_n, W_m)$  olan oranı şeklinde tanımlanmaktadır (Ozdemir vd. 2011).

$$R[W_{m+n-2}] = \frac{R[W_m] + R[W_n]}{P_s(W_m, W_n)}$$

Örneğin, temel yapıtaşı olarak kullanılan  $W_3$  ağının maliyeti 1 birim alındığında, iki adet  $W_3$  ağını birleştirerek oluşturulacak bir adet  $W_4$  ağının maliyeti, 4.5 birim bulunmaktadır. Yani her bir  $W_4$  ağını oluşturmak için ortalama 4.5 tane  $W_3$  ağı harcamak gerekmektedir. Çalışmamızda geliştirmeyi hedeflediğimiz kuantum devrelerdeki ilk amacımız, mevcut devreyi, birleştirme ihtimalini yükseltecek şekilde zenginleştirmektir. Böylece kaynak maliyetinin azaltılması sağlanacaktı. En iyi ihtimalle, birleştirme başarı olasılığını 1'e ulaştırabilen, “deterministik” kuantum devrenin geliştirilmesiyle, kaynak maliyeti parametremiz limit değerine ulaşmış olacaktır. Örneğin yukarıdaki hesap gözönünde bulundurulduğunda, her bir  $W_4$  ağını oluşturmak için, her seferinde yalnızca 2 tane  $W_3$  ağı harcamak kesinlikle yeterli olacaktır.



**Şekil 3.1:** Ozdemir ve diğ. (2011) devresinin çeşitli birleştirme senaryolarıyla kullanımının kaynak maliyet performansını görebilirsiniz.









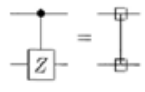



### 3.1.2. Birleştirilen Ağ Sayısı

Birleştirilen ağ sayısı büyük boyutta W ağı elde etmek üzere mevcut en iyi kuantum devre, yalnızca iki adet W ağını birleştirmektedir. Amacımız çok sayıda W ağını tek bir devrede bir seferde birleştirmeyi başarmak olduğundan, oluşturacağımız devrelerin birleştirebileceği W ağı sayısı, çalışmamızın ikinci öncelikli parametresi, birleştirilen ağ sayısıdır.

Kaynak maliyetinden verilecek ödünle, birleştirilen ağ sayısının arttırılabilmesi, dolayısıyla iki parametrenin birbiriyle olan ilişkisi de ayrıca incelenecektir.

### 3.2. YÖNTEM

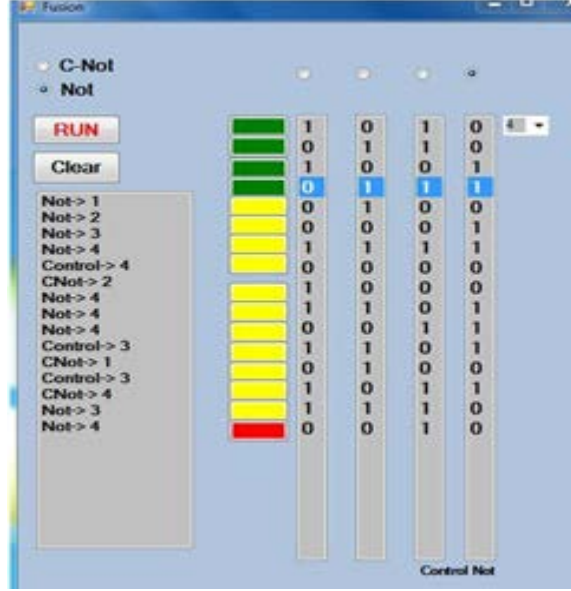
1-İlk olarak, kuantum optik devre elemanları, çalışmamızın amacı kapsamında incelenedik. Bu devre elemanlarının kuantum gösterimleri ve matematiksel modelleme matrisleri şu şekildedir:

Hadamard		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$		
Pauli-X		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$		
Pauli-Y		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$		
Pauli-Z		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$		
Phase		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$		
$\pi/8$		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$		
controlled-NOT			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	
swap			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	
controlled-Z			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$	
controlled-phase			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix}$	
Toffoli			$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	
Fredkin (controlled-swap)			$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	

Şekil. 3.2: Deneysel düzeneklerdeki devre çizimlerinde Pauli- X kapısı HWP, Hadamard kapısı QWP vs. Olarak gösterilmektedir

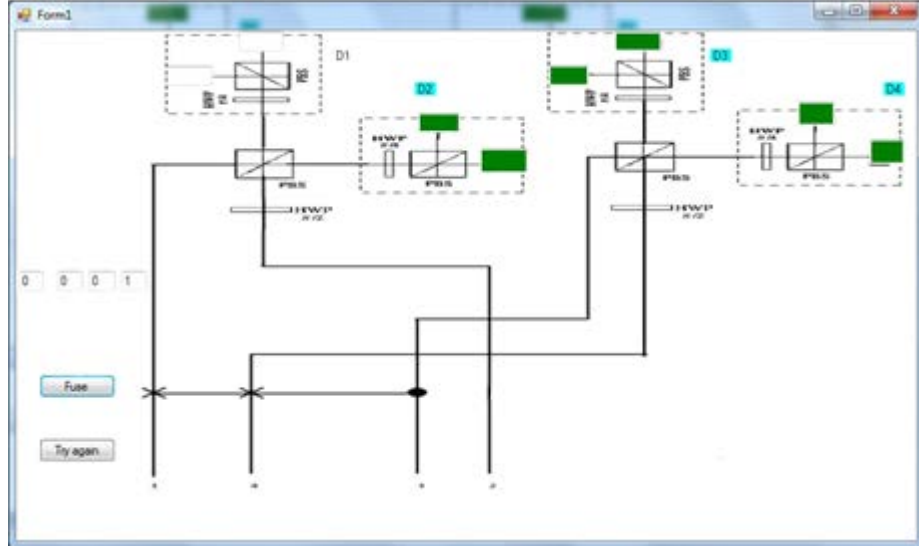
Temel kuantum devre elemanlarının her birinin, kombinasyonel olarak kullanımının, birleştirme ve genişletme amacımıza olacak potansiyel katkılarının bilgisayar simülasyonları ile analizi. Yöntemimizin bu fazı hakkında fikir vermesi açısından, ön hazırlık ol arak geliştirdiğimiz simülasyonlardan bazı ekran çıktılarını, aşağıdaki Şekillerde sunuyoruz..Birleştirme ve genişletme amacımıza olacak potansiyel katkılarının bilgisayar simülasyonları ile analizi. Yöntemimizin bu fazı hakkında fikir vermesi açısından, ön hazırlık ol arak geliştirdiğimiz simülasyonlardan bazı ekran çıktılarını, aşağıdaki Şekillerde sunuyoruz. Şekil 3.3'te, dört foton girişli, CNOT ve NOT kapıları içeren bir kuantum optik devresinde, kapıların hem sıralarının hem de

uygulandıkları fotonların dizilişlerinin denenmesiyle ortaya çıkan sonuçlar yer almaktadır.



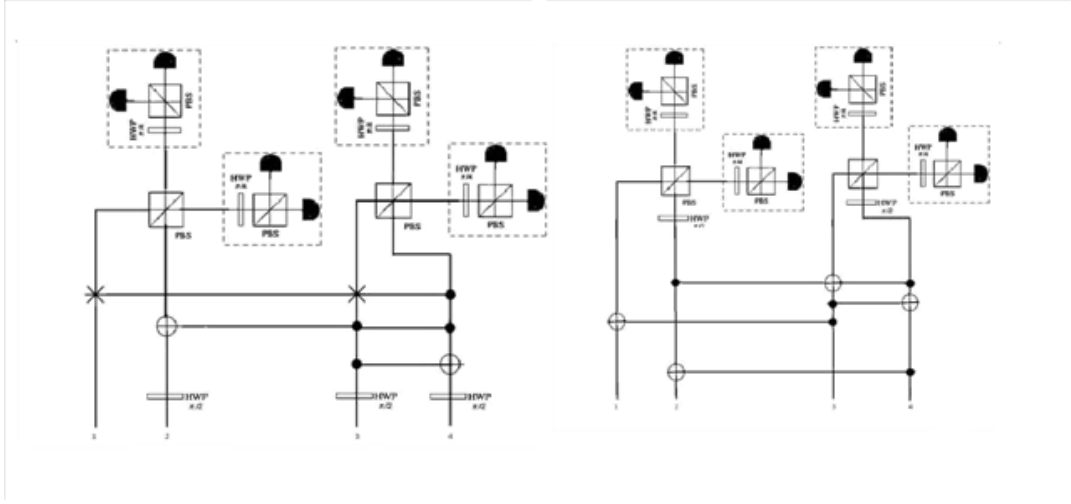
**Şekil 3.3:** Dört kuantum bitlik bir devrede, CNOT ve NOT (Pauli-X) kapılarının alternatif kullanım sonuçlarımız

2-Kurduğumuz modellere göre geliştireceğimiz kuantum devrelerin, W ağlarını birleştirme durum ve bu durumlara ait olasılıklarının bulunması için hazırladığımız kuantum simülasyon yöntemlerine dair örnek bir ekran çıktısı, Şekil 3.4'te sunulmuştur. Buradaki simülasyon, üç adet W ağını tek seferde birleştirme yöntemlerinin geliştirilebilmesine destek olmak üzere hazırlanmıştır. Sol taraftaki sıfır ve bir bitleri, birleştirilmesi hedeflenen ağlardan gelen birer fotonun, sırasıyla yatay ve dikey polarizasyonlarını simgelemektedir. Polarize hüzme ayırıcılar (PBS) ve foton dedektörleri, Ozdemir ve diğ. [28]'deki devreden alınmıştır. Bu devreye, Controlled Swap (soldaki iki fotonun polarizasyon durumuna göre üçüncü ve dördüncü fotonları kendi aralarında değiştiren) optik kapı entegre edilerek, birleşme durumları incelenmiştir. Verilen durumda, dört adet foton dedektöründen üç yalnızca tanesi, foton tespit ettiğinden, bu birleştirme denemesi başarısız olmuştur.

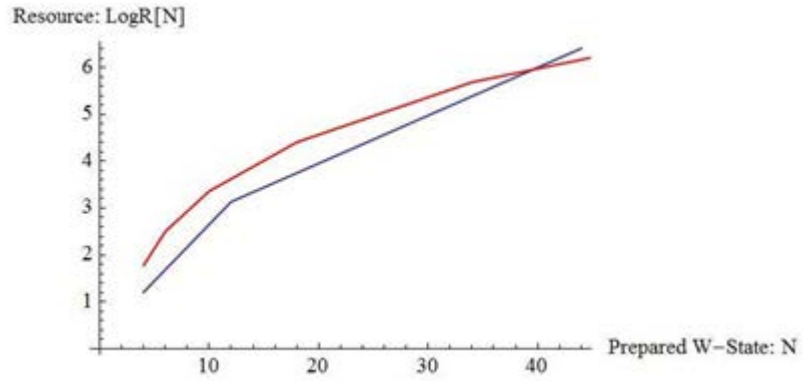


**Şekil 3.4:** İki kiden fazla W ağını birleştirebilmeyi denemek için hazırladığımız bir simülasyonun ekran çıktısı.

3-Başarılı devreler kurulduğunda, kullanılan her bir devre elemanı, devrenin kurulum ve çalışma maliyetini arttıracığından, (elektronik devrelerde Karno haritalarının kullanılmasını andıran) kuantum indirgeme yöntemleriyle, aynı işi görecekle daha küçük devrelerin bulunması aşamasına geçilecektir. Bu aşamada kullandığımız temel bir yöntem, Eisert tarafından geliştirilen, lineer optik devreler için konveks optimizasyon ve lineer programlamadır. Kuantum indirgeme yaparak, örneğin, Şekil 3.5(solda)'teki kuantum devreyi, Şekil 3.5(sağda)'teki kadar sadeleştirebilmekteyiz. Şekil 3.5(solda) 'teki HWP (Not) kapılarını, Şekil 3.5(sağda)'teki yöntemle ileri aşamaya kaydırıp, optimizasyon yoluna gittiğimizde; 1 CNOT, 1 Toffoli ve 1 Fredkin yerine Şekil 4b'deki gibi, 3 CNOT ve 1 Toffoli kapısıyla da aynı birleştirme işleminin gerçekleştirilebileceği bulunmaktadır. Deneysel gerçekleştirme ve cihaz haline getirip teknolojiye kazandırma süreci dikkate alındığında, (3 foton etkileşimi 2 foton etkileşiminden çok daha meşakkatli olduğundan) 2 adet CNOT kapısı, 1 adet Fredkin kapısından daha ucuz maliyete sahip olduğu düşünüldüğünde, sadeleştirme yöntemiyle, W haberleşme ağları oluşturma modellerimizi daha verimli hale getirebiliyoruz.



**Şekil 3.5:** (solda) Fotonik W ağlarını birleştirmek üzere geliştirdiğimiz bir kuantum optik devresi. (sağda) soldaki devrenin sadeleştirilmiş hali



**Şekil 3.6:** Mevcut devre (kırmızı eğri, Ozdemir ve diğ. 2011) ile deneme devremizin (mavi eğri) kaynak maliyet performansı performansı açısından karşılaştırılması.

4-Geliştirdiğimiz devrelerin sistemlerin kaynak maliyetleri açısından karşılaştırılıp, en verimli olanların tesbiti için, temel parametrelerimiz olan kaynak maliyeti ve tek seferde birleştirilen ağ sayısını kullandık. Mevcut en iyi yöntem olan Ozdemir ve diğ. [28] makalesindeki “mevcut devre” tek seferde iki adet W ağı birleştirdiğinden, aynı maliyete sahip de olsa tek seferde ikiden fazla ağı birleştirebilen devrelerimiz, bu parametre açısından başarılı sayılacaktır. Kaynak maliyeti açısından da, kaç adet ağı



birleştirdiğimizden bağımsız olarak, mevcut devreden daha düşük maliyetli devreler, yine başarılı sayılacaktır. Burada temel iki yaklaşım benimsemekteyiz:

- a) Herhangi bir hedef boyuttaki  $W$  ağını, mevcut devreden daha düşük bir kaynak maliyetiyle üretimini başarabilmek,
- b) Tüm hedef boyutlardaki  $W$  ağlarını, mevcut devreden daha düşük bir kaynak maliyetiyle üretimini başarabilmek.

Bu yaklaşımlardan ilkinine göre, çalışmanın ön hazırlık kapsamında çalıştığımız devrelerden birinin maliyeti (mavi eğri), hedef boyutu 40 civarı (yani 40 fotonluk)  $W$  haberleşme ağına kadar, mevcut devreden (kırmızı eğri) daha iyi çıktığını, Şekil 3.6'da görmek mümkündür.

### 3.3 SİMULASYONDA KULLANILAN PROGRAMLAR

Kuantum ağlerin oluşturulma ve genişletme maliyet hesaplarını yapmak ve bunları Monte Carlo simülasyonundaki(ihtimale dayalı yapılarda, olayı defalarca oynatıp ortalamayı alma) gibi simüle etmek ve grafiklerini çizdirmek için bazı programlama dilleri ve çeşitli araçlar kullandık. Bunlar C, C++, Python ve Wolfram Mathematica.

C, modern programlama dilleri arasında en hızlı çalışan dildir. Yapmış olduğumuz simülasyonların içeriğinde kurgulamış olduğumuz yapıyı binlerce kez çalıştırmamız gerekiyor. Bu hesaplamalardan bazıları 2-3 gün sürebiliyordu. C kullanarak bu süreyi minimuma indirdik. Ayrıca C'de bellek yönetimine müdahale olanağı diğer dillere göre çok daha fazla olduğu için bellek aşımı sorununu da en aza indirmiş olduk.

C++, nesne yönelimli program olduğu için yazması uzun, karmaşık olan ve kendi içinde defalarca tekrar eden fonksiyonları içeren maliyet hesaplamalarında yazım kolaylığı verdiği, hata alınması durumunda kod bloklarının kontrol edilmesi ve kodların debug





```

#####
import random, math, sys

# Up Limit: When state appears in that set
uplimit = 5
runtimes = 10

sys.setrecursionlimit(100000000)

#set of W states wrt sizes. ksi starts @ 0
WSet = [ [0,0], [0,0], [0,0], [0,0], [0,0], [0,0], [0,0], [0,0], [0,0] ]

# pointer for the current working set
ksi = 0

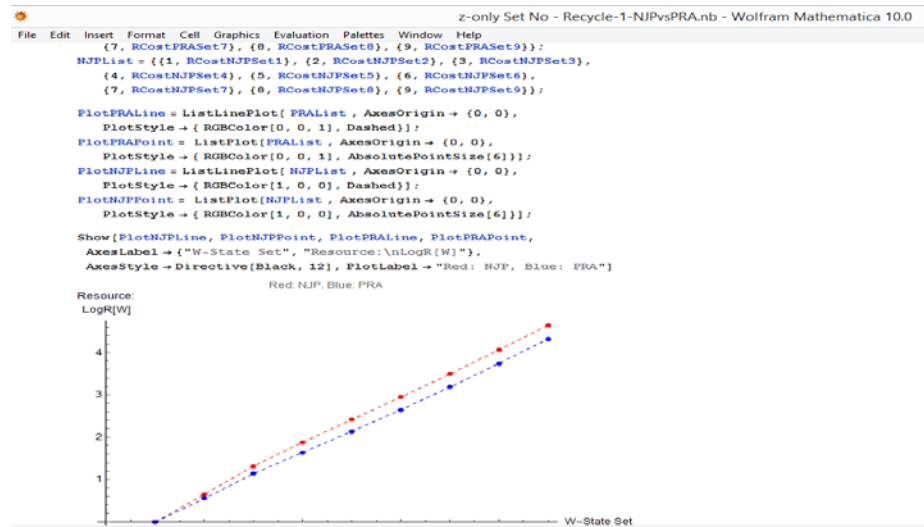
# Resource Cost
RCost = 0

def fuse(n,m):
    weighted_choices = [('Recycle', (n-1)*(m-1)), ('Success', n+m-2), ('Failure', 1)]
    population = [val for val, cnt in weighted_choices for i in range(cnt)]
    return random.choice(population)

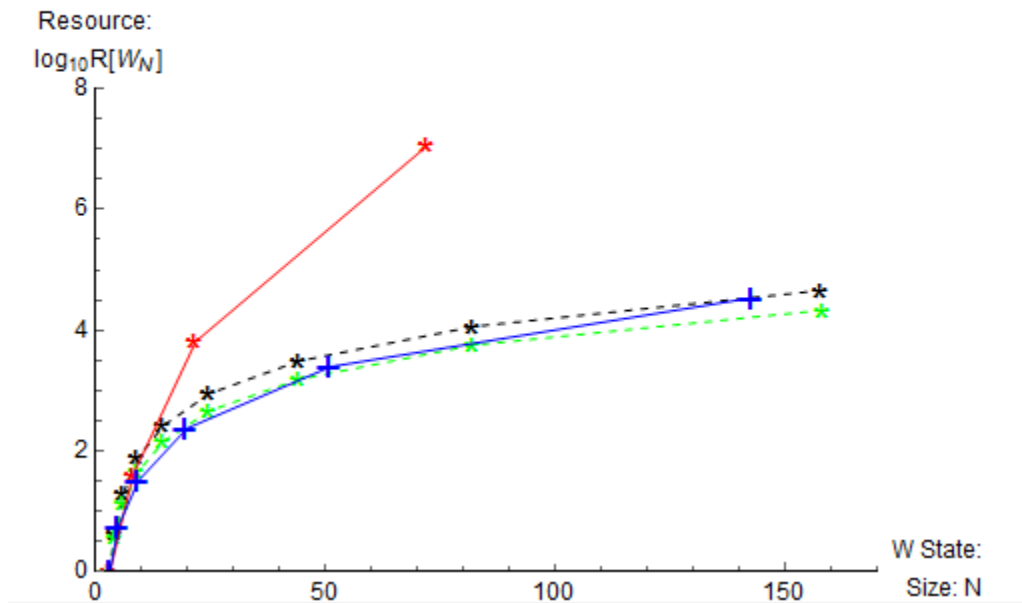
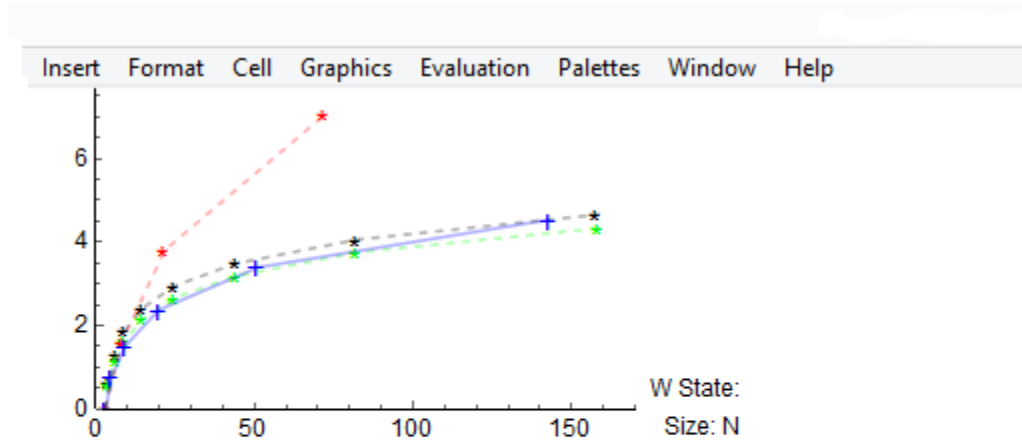
def placeNRec(size):
    Nksi = int(math.ceil( math.log(size-2,2) ))
    if WSet[Nksi][0]>0 and WSet[Nksi][1]>0:
        #print 'ERRROOOORR'
        sys.exit(1)
    elif WSet[Nksi][0]>0 and WSet[Nksi][1]==0:
        WSet[Nksi][1]=size
    elif WSet[Nksi][0]==0:
        WSet[Nksi][0]=size
    else:
        #print "ERRRROOOORRR --- 2 ---"
        sys.exit(1)
    #print "FROM REC", WSet

```

Şekil 3.9. un devamı



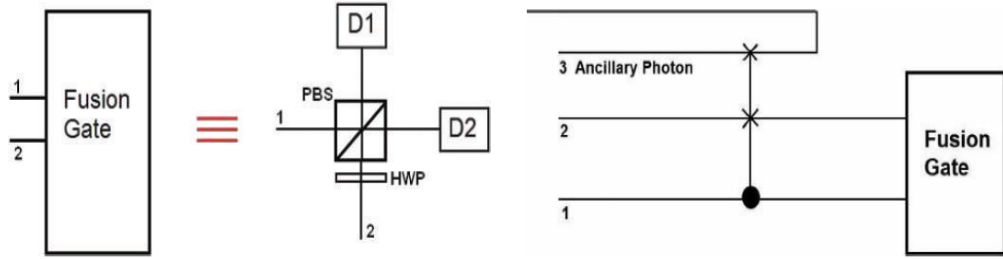
Şekil 3.10: Mathematica'da yazılmış simülasyon ve çizdirilen grafikler.



Şekil 3.10. devamı

## 4. BULGULAR

1- Çalışmalarımız sonucunda 1 adet ancillary(yardımcı) fotonu ve fredkin gate'i kullanarak şimdiye kadar ki en verimli olan özdemir vd. nin devresinin kuantum ağların birleşme olasılığını yüzde 50'den yüzde 75'e çıkaran devreyi tasarladık.

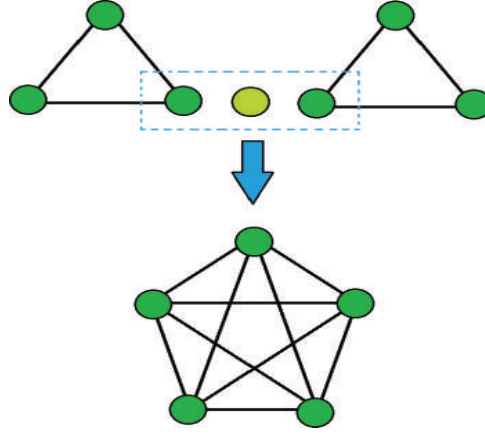


**Şekil 4.1:** Fredkin gate kullanarak tasarladığımız devre . Bugu ve diğ.(2013).

Böylelikle 2 tane 3 qubitlik W state ağı ve 1 adet H polarize ancillary fotonu geliştirmiş olduğumuz devrede kullanarak 5 qubitlik W state devresi elde ettik. Birleştirme başarı olasılığı arttığı için doğal olarak maliyet de buna ters oranda azaldı. Bu devrenin birleştirme başarı şansını arttırmasının dışında 2 önemli yeniliği var. İlki eski devrelerin tersine biz bununla 2 qubitlik W stateleri de birleştirebiliyoruz. Diğeri ise olasılık tablosunda failure'ı tamamen ortadan kaldırmış oluyoruz.

Input	Probability	FG	FG and F
$H, H$	$\frac{(n-1)(m-1)}{nm}$	recycle	recycle
$H, V$	$\frac{(n-1)}{nm}$	success	success
$V, H$	$\frac{(m-1)}{nm}$	success	success
$V, V$	$\frac{1}{nm}$	failure	success

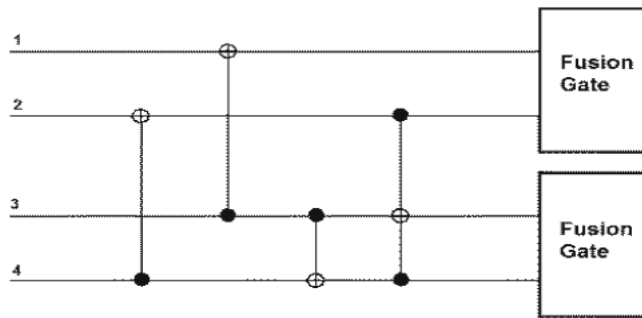
**Şekil 4.2:** Yeni devremizin başarı olasılık tablosu Bugu ve diğ.(2013).



**Şekil 4.3:** Yeni devremizin 3 qubitlik 2 W state'i (Bell state'leri) ancilla foton kullanarak (sarı renkli) birleştirmesini gösteriyor. Bugu ve diğ.(2013)

2-Haberleşmenin yalnızca 2 ağ arasında değil de 2'den fazla ağlar arasında yapılmasının önemli olduğunu bilinen bir gerçektir. Bu noktada 4 adet W state'i 2 tane fusion gate (FG) ve bir adate Toffoli gate kullanarak aynı anda birleştiren devre geliştirdik. Bu devreyle, klasik fusion gate ile büyük W state'ler oluşturmak istediğimizde ortaya çıkan üstsel maliyet sorununu da çözmüş olduk. Bu devrede başası oranı 4/16, başarısızlık oranı 11\16, geri dönüşüm (recycle) oranı ise 1\16 dır. Örneğin 4 adet 3 qubitlik W state'i (W3 'ün 1 tanesine maliyet açısından 1birim dersek) birleştirip 8 qubitlik W-8 elde etmenin maliyeti:

$$R[W-8] = R[W-3+3+3+3-4] = (R[W-3] + R[W-3] + R[W-3] + R[W-3]) / PS(W-3, W-3, W-3, W-3) = 40.5 \text{ br}$$



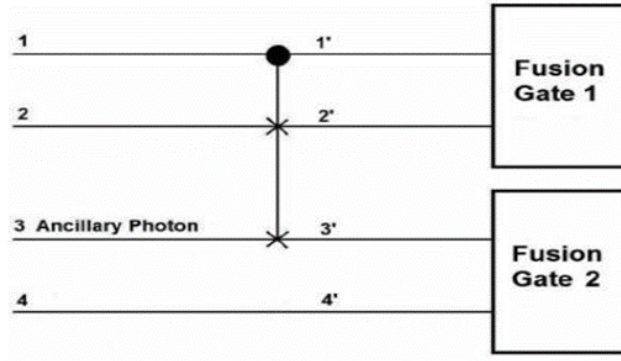
**Şekil 4.4:** Toffoli gate kullanarak geliştirdiğimiz devre Yesilyurt ve diğ. (2013)

Probability	Input	After 1st CNOT	After 2nd CNOT	After 3rd CNOT	After Toffoli	Result
$\frac{(n-1)(m-1)(t-1)(z-1)}{nm tz}$	0000	0000	0000	0000	0000	R
$\frac{(n-1)(m-1)(t-1)}{nm tz}$	0001	0101	0101	0101	0111	F
$\frac{(n-1)(m-1)(z-1)}{nm tz}$	0010	0010	1010	1011	1011	F
$\frac{(n-1)(m-1)}{nm tz}$	0011	0111	1111	1110	1110	F
$\frac{(n-1)(t-1)(z-1)}{nm tz}$	0100	0100	0100	0100	0100	F
$\frac{(n-1)(t-1)}{nm tz}$	0101	0001	0001	0001	0001	F
$\frac{(n-1)(z-1)}{nm tz}$	0110	0110	1110	1111	1101	F
$\frac{(n-1)}{nm tz}$	0111	0011	1011	1010	1010	S1
$\frac{(m-1)(t-1)(z-1)}{nm tz}$	1000	1000	1000	1000	1000	F
$\frac{(m-1)(t-1)}{nm tz}$	1001	1101	1101	1101	1111	F
$\frac{(m-1)(z-1)}{nm tz}$	1010	1010	0010	0011	0011	F
$\frac{(m-1)}{nm tz}$	1011	1111	0111	0110	0110	S2
$\frac{(t-1)(z-1)}{nm tz}$	1100	1100	1100	1100	1100	F
$\frac{(t-1)}{nm tz}$	1101	1001	1001	1001	1001	S3
$\frac{(z-1)}{nm tz}$	1110	1110	0110	0111	0101	S4
$\frac{1}{nm tz}$	1111	1011	0011	0010	0010	F

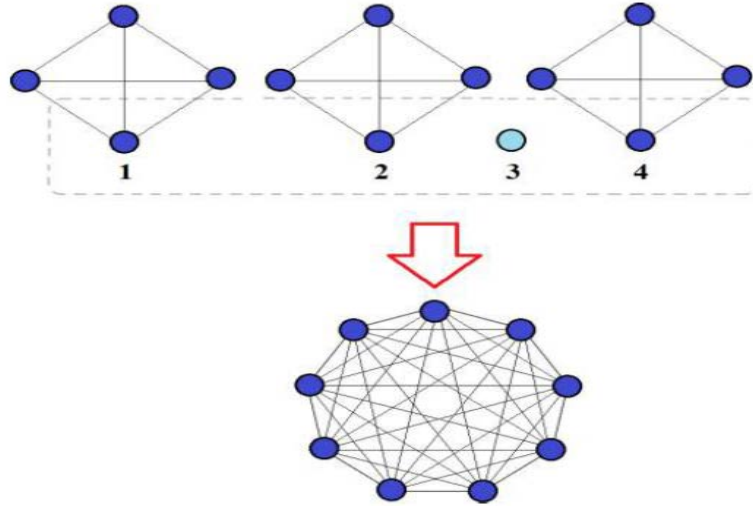
**Şekil 4.5:** Geliştirdiğimiz devrenin W state'leri birleştirme başarı olasılık tablosu Yesilyurt ve diğ. (2013)

3-Hem düşük maliyetli hem de aynı anda birden fazla W state'i birleştirip tek seferde daha büyük qübitlik W state'ler elde etmek bu alanda varılması gereken en önemli hedefti. Biz son olarak tam da bunu yapabilen bir devre geliştirdik Bu devre şimdiye kadar geliştirmiş olduğumuz devrelerin maliyet ve verimlilik açısından en iyisi. Bu devre hem Bell State'ler de dahil olmak üzere keyfi olarak verilen büyüklükteki W State'leri birleştirebiliyor hem de maliyet açısından bir önceki çalışmalara göre daha verimli. Bu devrede ilk defa Partial Recycle (Kısmi geri dönüşüm) terimini kullandık. Çünkü 2 adet W Fusion Gate olduğu ve devreye 2 adet W state girdiği için bunlardan biri birleşme bile diğerinin ancilla foton sayesinde Failure olmama (yani ziyan olmama) durumu var. vrede ilk defa Partial Recycle (Kısmi geri dönüşüm) terimini kullandık. Çünkü 2 adet W Fusion Gate olduğu ve devreye 2 adet W state girdiği için bunlardan biri birleşme bile diğerinin ancilla foton sayesinde Failure olmama (yani ziyan olmama) durumu var.





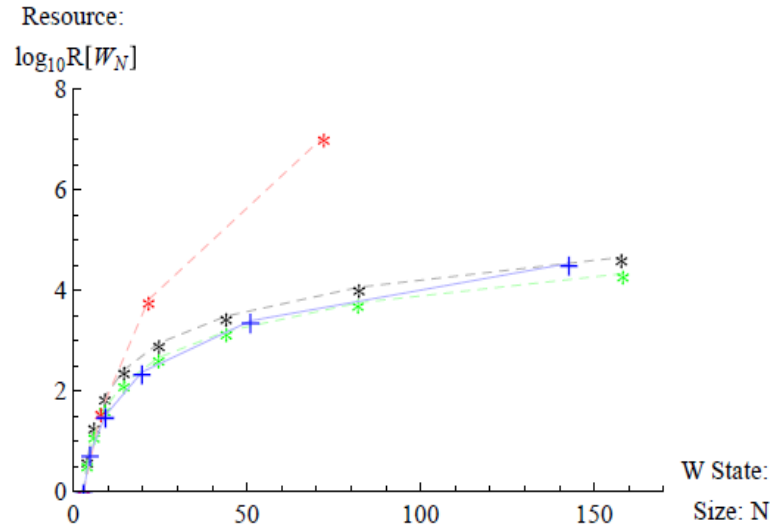
Şekil 4.6: Ancilla foton ve Fredkin gate kullanarak geliştirdiğimiz Fusion(birleştirme) devresi.



Şekil 4.7: Fredkin gate ve ancilla foton kullanarak geliştirdiğimiz devrenin tek seferde 3 adet  $W_4$  kullanarak  $W_9$  elde etmesini gösteren şekil. Ozaydin ve diğ.(2014)

Probability	Input	Throughput	FG1, FG2	Result
$\frac{(n-1)(m-1)(t-1)}{nm}$	$H, H, H, H$	$\{H, H\}, \{H, H\}$	$R, R$	$R$
$\frac{(n-1)(m-1)}{nm}$	$H, H, H, V$	$\{H, H\}, \{H, V\}$	$R, S$	$PR$
$\frac{(n-1)(t-1)}{nm}$	$H, V, H, H$	$\{H, V\}, \{H, H\}$	$S, R$	$PS$
$\frac{(n-1)}{nm}$	$H, V, H, V$	$\{H, V\}, \{H, V\}$	$S, S$	$S1$
$\frac{(m-1)(t-1)}{nm}$	$V, H, H, H$	$\{V, H\}, \{H, H\}$	$S, R$	$PS$
$\frac{(m-1)}{nm}$	$V, H, H, V$	$\{V, H\}, \{H, V\}$	$S, S$	$S2$
$\frac{(t-1)}{nm}$	$V, V, H, H$	$\{V, H\}, \{V, H\}$	$S, S$	$S3$
$\frac{1}{nm}$	$V, V, H, V$	$\{V, H\}, \{V, V\}$	$S, F$	$F$

Şekil 4.8: Geliştirmiş olduğumuz devrenin başarı olasılık tablosu. Ozaydin ve diğ. (2014)



**Şekil 4.9:** Grafikte, Kırmızı renk (Can ve diğ. 2013) in, mavi renk son olarak sunmuş olduğumuz devrenin, siyah renk Özdemir ve diğ. (2011) in, yeşil renk ilk bulgumuz olan Bugu ve diğ. (2013)'ün maliyetini gösteriyor. Grafik için 1000 adet birleştirme denemesi yapıp ortalamaları alınmıştır. Ozaydin ve diğ.(2014)

## 5. TARTIŞMA VE SONUÇ

Çalışmalarımız sonucunda kuantum haberleşme için gerekli olan büyük newtworkler elde etmek için gerekli olan ve Bell state de dahil olmak üzere keyfi sayıdaki qubitlere sahip W state kuantum ağlarını birleştiren ve/veya genişleten devreler geliştirdik. Bu çalışma, bu konularda daha önce çalışan grupların elde ettikleri sonuçlardan maliyet ve başarı olasılığı açısından daha verimlidir. Geliştirmiş olduğumuz devrelerimizde Toffoli gate, Fredkin gate ve ancilla kullanmış olmamız çalışmalarımızın en büyük yeniliklerindedir. Bir başka yenilik de, maliyet hesabını direkt olarak etkileyen kısmi başarı(partial success), kısmi geri dönüşüm(partial recycle) gibi terimleri bu alana kazandırmış olmamız. Bu çalışma kapsamında yazmış olduğumuz 3 tane makale dergilerde yayımlandı[5,24,38].

Yapmış olduğumuz bu teorik çalışmalar günümüz fotonik teknolojiyle yapılabilir durumdadırlar. Bunların yapılması durumunda kuantum haberleşme, kuantum hesaplama ve kuantum şifreleme gibi bilim alanlarına ciddi katkılar sağlayabilir. Biz her ne kadar ağların genişletilmesindeki başarı oranını yükseltsek de henüz deterministik bir çözüme ulaşamadık. Bu konuda büyük bir açık var. Deterministik çözüme ulaşmak ayrı bir çalışma, deterministik (yüzde yüz başarı oranıyla birleştirme) çözüme kuantum logic kapılarını uygulanabilir ve çok fazla sayıda olmamak şartıyla ulaşabilmek ayrı bir çalışma olarak yapılabilir. Bu konularda yapılacak çalışmalar önemli dergilerde yayımlanır. Ayrıca yine W state'lerin etkileşimleri incelenerek, ilk zamanlarda iktisat biliminde kullanılan Fisher information kullanılarak, farklı durumdaki W state'lerin arasındaki fazlar ölçülebilir ve bu ölçümler W, GHZ ve Dicke state'ler arasında farklı kombinasyonlarla incelenebilir.

Teknolojinin ilerlemesiyle klasik bilgisayarlar, mevcut işlemci mimarileri yüzünden önemini kaybetmeye başladılar. Bu durum bilim adamlarının kuantum temelli bilgisayar teknolojilerine daha fazla ağırlık vermesine neden oldu. Efektif olarak kullanılacak kuantum bilgisayarlarıyla hiçbir şekilde kırılmayacak şifreli

haberleşme yapılabilecek, klasik bilgisayarlarla hesaplaması uzun sürebilecek işlemler (örneğin görüntü bazlı sınıflandırma[17] gibi işlemler) daha kısa sürede yapılabilir hale gelecek ve bunun gibi çok daha faydalı işlerde kullanılabilir. Bu yüzden kuantum bilgisayarının yapılması ve kullanılabilmesi için gerekli olan kuantum haberleşme, kuantum bilgi teknolojisi, spintronics ve nano-electronics gibi alanlarda yapılacak gelişmelerden her biri çok önemlidir.

## KAYNAKLAR

- [1]. Altintas A. A., Ozaydin F., Yesilyurt C., Bugu S.,2014, Constructing quantum logic gates using q-deformed harmonic oscillator algebras, *Quantum Information Processing* 13 (4) 1035-1044.
- [2]. Bell J. S. , 1964 ,On the Einstein Podolsky Rosen paradox, *Physics* 1, 3, 195–200.
- [3]. Bell J. S., 1966, On the Problem of Hidden Variables in Quantum Mechanics *Review of Modern Physics* 38, 447.
- [4]. Browne D. E., Rudolph T., 2005, Resource-Efficient Linear Optical Quantum Computation *Physical Review Letters* 95 010501.
- [5]. Bugu S., Yesilyurt C., Ozaydin F.,2013, Enhancing the W-state quantum-network-fusion process with a single Fredkin gate, *Physical Review A*, 87, 032331.
- [6]. Deutsch D.,1985, Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer, *Proceeding of the Royal Society of London A* 400 97.
- [7]. Einstein A., B. Podolsky, and N. Rosen, 1935, Can Quantum - Mechanical Description of Physical Reality Be Considered Complete?, *Physical Review* 47, 777
- [8]. Ekert A. K., 1991, Quantum cryptography based on Bell's theorem,*Physical Review Letters* 67,661.
- [9]. Ellie D., Prakash P., 2006, The computational power of the W and GHZ states, *Quantum Information Computation* Vol. 6, pp.173-183.
- [10]. Erol V., Bugu S., Ozaydin F., Altintas A. A., *Signal Processing and Communications Applications Conference (SIU)*,2014
- [11]. Erol V., Ozaydin F., Altintas A.A., 2014, Analysis of Entanglement Measures and LOCC Maximized Quantum Fisher Information of General Two Qubit Systems, *Scientific Reports* 4.
- [12]. Erol V., 2015, A Comparative Study of Concurrence and Negativity of General three-level quantum systems of two particles, *AIP Conference Proceeding* 1653, 020037.
- [13]. Feynman R.P. ,1982, Simulating Physics with Computers, *International Journal of Theoretical Physics* VoL 21, Nos. 6/7.

- [14]. Fiurasek J., 2006, Linear-optics quantum Toffoli and Fredkin gates, *Physical Review A* 73, 062313.
- [15]. Fiurasek J., 2008, Linear Optical Fredkin gate based on partial-SWAP gate, *Physical Review A* 78, 032317.
- [16]. Haffner H., 2005, Scalable multiparticle entanglement of trapped ions, *Nature* 438 643.
- [17]. Gorgel P., Sertbas A., Ucan O.N., (2014). Computer-aided classification of breast masses in mammogram images based on spherical wavelet transform and support vector machines, *Expert Systems*, (DOI: 10.1111/exsy.12073).
- [18]. Leibfried D., 2005, Creation of a six-atom 'Schrödinger cat' state, *Nature* 438 639.  
Li Y., Kobayashi T., 2004, Four-photon W state using two-crystal geometry parametric down-conversion, *Physical Review A* 70 014301.
- [19]. Liu Y. X., Miranowicz A., Koashi M., Imoto N., 2002, Realization of symmetric sharing of entanglement in semiconductor microcrystallites coupled by a cavity field *Phys. Rev. A* 66 062309.
- [20]. Mikami H., Li Y., Fukuoka K., Kobayashi T., 2005, New High-Efficiency Source of a Three-Photon W State and its Full Characterization Using Quantum State Tomography, *Physical Review Letters* 95 150404.
- [21]. Ozaydin F., Altintas A.A., Bugu S., Yesilyurt C., Arik M., 2014, Quantum Fisher Information of Several Qubits in the Superposition of A GHZ and two W States with Arbitrary Relative Phase, *International Journal of Theoretical Physics* 53 (9), 3219- 3225.
- [22]. Ozaydin F., Altintas A. A., Bugu S., Yesilyurt C., 2013, Quantum Fisher Information of N Particles in the Superposition of W and GHZ States, *International Journal of Theoretical Physics* 52 (9), 2977-2980.
- [23]. Ozaydin F., Altintas A.A., Bugu S., Yesilyurt C., 2014, Alpha Head on Collision with a Fixed Gold Nucleus, Taking into Account the Relativistic Rest Mass Variation as Implied by Mass-Energy Equivalence, *Acta Physica Polonica A* 125 (2), 618-619.
- [24]. Ozaydin F., Bugu S., Yesilyurt C., Altintas A.A., Tame M., Ozdemir S.K., 2014, Fusing multiple W states simultaneously with a Fredkin gate, *Physical Review A* 89(4), 042311.
- [25]. Ozaydin F., 2014, Quantum Fisher Information of W States in Decoherence Channels, *arXiv Preprint arXiv:1403.2376*
- [26]. Ozaydin F., 2015, Quantum Fisher Information of a  $3 \times 3$  Bound Entangled State and its Relation with Geometric Discord, *International Journal of Theoretical Physics*, 1-7.

- [27]. Ozayin F.,2014, Phase damping destroys quantum Fisher information of W states, *Physics Letters A* 378 (43), 3161-3164.
- [28]. Ozdemir S. K., Matsunaga E., Tashima T., Yamamoto T., Koashi M., Imoto N.,2011, An optical fusion gate for W-states, *New Journal of Physics* 13, 103003.
- [29]. Resch K. J., Walther P. Zeilinger A.,2005, Full Characterization of a Three-Photon Greenberger-Horne-Zeilinger State Using Quantum State Tomography, *Physical Review Letters* 94 070402.
- [30]. Sharma S., Almeida E., Sharma N. K., 2008, Multipartite entanglement of three trapped ions in a cavity and W-state generation, *Journal of Physics B: Atomic, Molecular and Optical Physics* 41 165503.
- [31]. Shi B.S., Tomita A.,2005, Photonic Realization of Quantum Information Systems, *Journal Modern Optics* 52 755, (2005).
- [32]. Tashima T., Kitano T., Ozdemir S. K., Yamamoto T., Koashi M., Imoto N.,2010, Demonstration of Local Expansion Toward Large-Scale Entangled Webs, *Physical Review Letters* 105 210503.
- [33]. Tashima T., Ozdemir S. K., Yamamoto T., Koashi M., Imoto N.,2008, Elementary optical gate for expanding an entanglement web, *Phys. Rev. A* 77 030302.
- [34]. Tashima T., Ozdemir S. K., Yamamoto T., Koashi M., Imoto N.,2009, Local expansion of photonic W state using a polarization-dependent beamsplitter, *New Journal of Physics* 11 023024.
- [35]. Tashima T., Wakatsuki T., Ozdemir S. K., Yamamoto T., Koashi M., Imoto N.,2009, Local Transformation of Two Einstein-Podolsky-Rosen Photon Pairs into a Three- Photon W State, *Physical Review Letters* 102 130502.
- [36]. Teklemariam G., 2002, Quantum erasers and probing classifications of entanglement via nuclear magnetic resonance, *Physical Review A* 66 012309.
- [37]. Yamamoto T., Tamaki K., Koashi M. Imoto N.,2002, Polarization-entangled W state using parametric down-conversion, *Physical Review A* 66 064301.
- [38]. Yesilyurt C., Bugu S., Ozaydin F., 2013, An optical gate for simultaneous fusion of four photonic W or Bell states, *Quantum Information Processing* 12 (9),2965-2975 (2013)
- [39]. Yesilyurt C., Tan S. G., Liang G., Jalil M. B. A.,2015, Efficient Dual Spin-Valley Filter In Strained Silicene, *arXiv:1504.01969* .
- [40]. Zeilinger A., Horne M. A., Weinfurter H., Zukowski M.,1997, Three-Particle Entanglements from Two Entangled Pairs, *Physical Review Letters* 78 3031.

- [41]. Zou X., Pahlke K., Mathis W., 2002, Generation of an entangled four-photon W state, *Physical Review A* 66 044302.
- [42]. ([http://en.wikipedia.org/wiki/Timeline\\_of\\_quantum\\_computing/](http://en.wikipedia.org/wiki/Timeline_of_quantum_computing/)) [Ziyaret Tarihi: 10 Mayıs 2014].
- [43]. <http://www.nas.nasa.gov/quantum/>) [Ziyaret Tarihi: 10 Mayıs 2014].
- [44]. [www.technologyreview.com/news/537041/ibm-shows-off-a-quantum-computer-chip/](http://www.technologyreview.com/news/537041/ibm-shows-off-a-quantum-computer-chip/) [Ziyaret Tarihi: 10 Mayıs 2014]



## EKLER

**EK 1.** Simülasyonlarda ve grafik çizimlerinde kullandığımız programlama,script dilleri ve yardımcı araçlara ait bazı kod ve script örnekleri:

1.C++ :

```
#include <stdio.h>
#include <string>
#include <iostream>
#include <math.h>
#include <Windows.h>
#include "share.h"
#define ROW 7
#define COL 3
#define LTimes 1000 //10000

using namespace std;

//FUNCTION PROTOTYPES
char Fuse(int n, int m, int t);
void ClearMatrixRow(int matrix[ROW][COL], int row);
void ClearMatrix(int matrix[ROW][COL], int row, int col);
void PrintMatrix(int matrix[ROW][COL], int row, int col);
void PlaceN(int matrix[ROW][COL], int size, int *ksi);
void Step2(int matrix[ROW][COL], int *ksi, long long int *RCost, int *UpLimit);
void Step3(int matrix[ROW][COL], int *ksi, long long int *RCost, int *UpLimit);

//MAIN FUNCTION START
int main (void)
{
    double x;
    string fusionresult;
    int i,j,counter; int finalcounter=0;
    int WSet[ROW][COL];
    long long int RCost = 0;
    long long int WSize = 0;
    int ksi=0;
    int UpLimit = 6; //FINAL STATE NU
    int fuseeach=0;
    int fusetotal=0;

    ClearMatrix(WSet,ROW,COL);
    std::srand( std::time( NULL ) ); // seed random
    std::rand(); // throw out first value...

    PrintMatrix(WSet,ROW,COL);

    cout << " PROGRAM START" << endl;

    for(counter=1;counter<LTimes+1;counter++ ){
```

```

        ksi = 0;
        ClearMatrix(WSet,ROW,COL);
        Step2(WSet,&ksi,&RCost,&UpLimit);
        WSize = WSize + WSet[UpLimit-1][0];
        // cout << "Created W Size: " << WSet[UpLimit][0] << endl;
        finalcounter=finalcounter+1;
    }
    //////////////////////////////////////
    cout << "Counter : " << finalcounter << "\n RCOST: " << RCost << "\n Total W
Size: " << WSize << endl;
    PrintMatrix(WSet,ROW,COL);
        cout << " KSI : " << ksi << " UpLimit : " << UpLimit << endl;
        // cout
    cout << " PROGRAM OVER PROGRAM OVER PROGRAM OVER PROGRAM OVER" << endl;
    getchar();

    return 0;

}
//MAIN FUNCTION OVER

////////////////////////////////////
//          FUNCTIONS          //////////////////////////////////
////////////////////////////////////

////////////////////////////////////
///          STEP 2          //////////////////////////////////
void Step2(int matrix[ROW][COL], int *ksi, long long int *RCost, int *UpLimit){
    if(*ksi==0){
        if(matrix[*ksi][0]==0){
            matrix[*ksi][0]=3;
            ++*RCost;
            Step2(matrix,ksi,RCost,UpLimit);
        }
        else if(matrix[*ksi][1]==0){
            matrix[*ksi][1]=3;
            ++*RCost;
            Step2(matrix,ksi,RCost,UpLimit);
        }
        else if(matrix[*ksi][2]==0){
            matrix[*ksi][2]=3;
            ++*RCost;
            Step2(matrix,ksi,RCost,UpLimit);
        }
        else{
            Step3(matrix,ksi,RCost,UpLimit);
        }
    }
    else{
        if(matrix[*ksi][0]==0 || matrix[*ksi][1]==0 || matrix[*ksi][2]==0){
            *ksi = *ksi - 1 ;
            Step2(matrix,ksi,RCost,UpLimit);
        }
        else{
            Step3(matrix,ksi,RCost,UpLimit);
        }
    }
}
////////////////////////////////////

```

```

////////////////////////////////////
//          STEP 3          /// ///
void Step3(int matrix[ROW][COL], int *ksi, long long int *RCost, int *UpLimit){
    int n,m,t;
    char FR;

    n=matrix[*ksi][0];
    m=matrix[*ksi][1];
    t=matrix[*ksi][2];

    ClearMatrixRow(matrix,*ksi);

    FR = Fuse(m,n,t);
    // A: Rec, B: PartRec, C: PartSuc, D: Suc, E: Fail
    if(FR=='E'){
        Step2(matrix,ksi,RCost,UpLimit);          // ksi *
    }
    else if(FR=='A'){
        // cout << "\t ***** A: RECYCLE *** " << endl;
        //Sleep(2000);
        if(n>3){ PlaceN(matrix,n-1,ksi); }          // ksi *
        if(m>3){ PlaceN(matrix,m-1,ksi); }          // ksi *
        if(t>3){ PlaceN(matrix,t-1,ksi); }          // ksi *
        Step2(matrix,ksi,RCost,UpLimit);          // ksi *
    }
    else if(FR=='B'){
        // cout << "\t ***** B: PARTIAL RECYCLE *** " << endl;
        // Sleep(2000);
        if(n>3){ PlaceN(matrix,n-1,ksi); }          // ksi *
        if(m>3){ PlaceN(matrix,m-1,ksi); }          // ksi *
        Step2(matrix,ksi,RCost,UpLimit);          // ksi *
    }
    else if(FR=='C'){
        // cout << "\t ***** C-111: PARTIAL SUCCESS *** " << endl;
        // Sleep(2000);
        if(t>3){ PlaceN(matrix,t-1,ksi); }          // ksi *
        PlaceN(matrix,n+m-2,ksi);                  // ksi *
        if(*ksi<*UpLimit - 1){
            Step2(matrix,ksi,RCost,UpLimit);          //
ksi *
        }
    }
    else if(FR=='D'){
        // cout << "\t ***** D-111: SUCCESS *** " << endl;
        // Sleep(2000);
        PlaceN(matrix,n+m+t-3,ksi);                // ksi *
        if(*ksi<*UpLimit - 1){
            Step2(matrix,ksi,RCost,UpLimit);          //
ksi *
        }
    }
    else{
        cout << "Unexpected Return From Fuse @2" <<
endl;
    }

    // cout << "THIS IS THE FINAL COUNTDOWN" << endl;
}
////////////////////////////////////

```

```

////////////////////////////////////
//          FUSE          //////////
char Fuse(int n, int m, int t){
    int fusion, PRec, PPartRec, PPartSuc, PSuc, PF;
    int limRec1, limRec2, limPartRec1, limPartRec2;
    int limPartSuc1, limPartSuc2, limSuc1, limSuc2;
    char thefusionResult;
    PRec = (n-1)*(m-1)*(t-1);
        limRec1 = 0;
        limRec2 = PRec+1;
    PPartRec = (n-1)*(m-1);
        limPartRec1 = PRec;
        limPartRec2 = PRec+PPartRec+1;
    PPartSuc = (n-1)*(t-1) + (m-1)*(t-1);
        limPartSuc1 = PRec+PPartRec;
        limPartSuc2 = PRec+PPartRec+PPartSuc+1;
    PSuc = m+n+t-3;
        limSuc1 = PRec+PPartRec+PPartSuc;
        limSuc2 = PRec+PPartRec+PPartSuc + PSuc + 1;
    PF = n*m*t - ( PRec + PPartRec + PPartSuc + PSuc );

    fusion = (rand() / ((double)RAND_MAX)) *(n*m*t-1) +1;
    if      (fusion>limRec1      && fusion < limRec2)      { thefusionResult = 'A';
}
    else if(fusion>limPartRec1 && fusion < limPartRec2) { thefusionResult = 'B';
}
    else if(fusion>limPartSuc1 && fusion < limPartSuc2) { thefusionResult = 'C';
}
    else if(fusion>limSuc1     && fusion < limSuc2)     { thefusionResult = 'D';
}
    else
}
}
return(thefusionResult);
}
////////////////////////////////////

////////////////////////////////////
//    PLACE whatever it is    //////////
void PlaceN(int matrix[ROW][COL], int size, int *ksi){

    int Nksi;
        // cout << "PlaceN will start with Size: " << size << endl;
    //////////////////////////////////////
        //Sleep(2000);
    //////////////////////////////////////
    //    PrintMatrix(matrix,ROW,COL);
    if(size==3)      { Nksi=0;  }
    else if(size>3  && size<7){ Nksi=1;  }
    else if(size>6  && size<16){ Nksi=2;  }
    else if(size>15 && size<43){ Nksi=3;  }
    else if(size>42 && size<124){ Nksi=4;  }
    else if(size>123 && size<372){ Nksi=5;  }
    else if(size>371 ){ Nksi=6;  }
    else { cout << "xx ACHTUNG @ PlaceN ACHTUNG!!! @ PlaceN ! ACHTUNG!!! @
PlaceNRec " << endl; }

    if(Nksi>*ksi){ *ksi=*ksi+1; ***** NEW KSI : " << *ksi
<<endl;
}
}
// cout << "@@@@PlaceN, New Ksi: " << Nksi << endl;

```

```

        if(matrix[Nksi][0]==0){
            matrix[Nksi][0]=size;
        }
        else if(matrix[Nksi][1]==0){
            matrix[Nksi][1]=size;
        }
        else if(matrix[Nksi][2]==0){
            matrix[Nksi][2]=size;
        }
        else{
            //Sleep(500);
            // cout << "Size: " << size << endl;
            // PrintMatrix(matrix,ROW,COL);
            // cout << " yyyyyyyyyyyyyyyy ACHTUNG !!! @@@@ PlaceN
ACHTUNG !!! @@@@ RECYCLE" << endl;
        }
    }
}
////////////////////////////////////////////////////

// CLEAR MATRIX ROW
void ClearMatrixRow(int matrix[ROW][COL], int row){
    int j;
    for(j=0;j<COL;j++){
        matrix[row][j]=0;
    }
}
////////////////////////////////////////////////////

// CLEAR MATRIX
void ClearMatrix(int matrix[ROW][COL], int row, int col){
    int i,j;
    for(i=0;i<row;i++){
        for(j=0;j<col;j++){
            matrix[i][j]=0;
        }
    }
}
////////////////////////////////////////////////////

// PRINT MATRIX
void PrintMatrix(int matrix[ROW][COL], int row, int col){
    int i,j;
    for(i=0;i<row;i++){
        cout << "{";
        for(j=0;j<col;j++){
            cout << matrix[i][j] << ",";
        }
        cout << "},";
    }
    cout << "]" << endl;
}
////////////////////////////////////////////////////

```

2. Python:

```
#####
#####
### (R) Sinan Buđu 2014.11.18 #####
#####
### PRA Fusing Similar sizes with recycle ###
#####
```

```
import random, math, sys
```

```
sys.setrecursionlimit(10000)
```

```
#set of W states wrt sizes. ksi starts @ 0
```

```
WSet = [ [0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0] ]
```

```
# pointer for the current working set
```

```
ksi = 0
```

```
# Resource Cost
```

```
RCost = 0
```

```
# Up Limit: When state appears in that set
```

```
uplimit = 5
```

```
def fuse(n,m):
```

```
    weighted_choices = [('Recycle', (n-1)*(m-1)), ('Success', n+m-1)]
```

```
    population = [val for val, cnt in weighted_choices for i in range(cnt)]
```

```
    return random.choice(population)
```

```
def placeNRec(size):
```

```
    Nksi = int(math.ceil( math.log(size-2,2) ))
```

```

if WSet[Nksi][0]>0 and WSet[Nksi][1]>0:
    #print 'ERRROOOORR'
    sys.exit(1)
elif WSet[Nksi][0]>0 and WSet[Nksi][1]==0:
    WSet[Nksi][1]=size
elif WSet[Nksi][0]==0:
    WSet[Nksi][0]=size
else:
    #print "ERRRROOOORRR --- 2 ---"
    sys.exit(1)
#print "FROM REC", WSet

def placeNSuc(size):
    global ksi
    ksi = ksi+1
    if WSet[ksi][0]>0 and WSet[ksi][1]>0:
        #print "ERRRROOOORRR --- 3.1 ---"
        sys.exit(1)
        #print "ERRRROOOORRR --- 3.2 ---"
    elif WSet[ksi][0]>0 and WSet[ksi][1]==0:
        WSet[ksi][1]=size
    elif WSet[ksi][0]==0:
        WSet[ksi][0]=size
    else:
        #print "ERRRROOOORRR --- 4 ---"
        sys.exit(1)
    #print "FROM SUC", WSet

def step2():

```

```

global ksi,RCost
if ksi==0:
    if WSet[ksi][0]==0 and WSet[ksi][1]==0:
        WSet[ksi][0]=3
        #print WSet
        RCost=RCost + 1
        step2()
    elif WSet[ksi][0]!=0 and WSet[ksi][1]==0:
        WSet[ksi][1]=3
        RCost=RCost + 1
        #print WSet
        # print "b"
        step2()
    elif WSet[ksi][0]!=0 and WSet[ksi][1]!=0:
        #print "calling STEP3 from 1"
        step3()
elif ksi>0:
    if WSet[ksi][0]==0 or WSet[ksi][1]==0:
        ksi = ksi-1
        #print "d"
        step2()
    else:
        #print "calling STEP3 from 2"
        step3()

def step3():
    global ksi, uplimit, FinalCost, RCost
    #print WSet
    n=WSet[ksi][0]
    m=WSet[ksi][1]

```



```

WSet[ksi][0]=0
WSet[ksi][1]=0
#print WSet

FR = fuse(n,m)
#print FR
#print RCost
if FR=='Failure':
    step2()
elif FR=='Recycle':
    if n>3: placeNRec(n-1)
    if m>3: placeNRec(m-1)
    step2()
elif FR=='Success':
    if ksi<uplimit-1:
        placeNSuc(n+m-2) #THIS SHOULD HAVE BEEN N+M-1
        step2()
    else:
        placeNSuc(n+m-2) #THIS SHOULD HAVE BEEN N+M-1
        FinalCost = RCost
        #print WSet
        #print "Reached Set: ", ksi
        #print "Final Log Cost ", math.log10(FinalCost)
    else:
        pass
        #print "Bir ihtimal daha var"

for i in range(110):
    ksi=0
    WSet = [ [0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0] ]

```

```

step2()

print (i, "kez calisti")
print("Total Final Cost=", FinalCost)
print("Average Cost=", FinalCost/i)
print(math.log10(FinalCost/i))
print(WSet)

#####
#####
### (R) Sinan Buğu #####
#####
### NJP Fusing Similar sizes with recycle ###
#####

import random, math, sys

# Up Limit: When state appears in that set
uplimit = 5
runtimes = 10

sys.setrecursionlimit(100000000)

#set of W states wrt sizes. ksi starts @ 0
WSet = [ [0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0] ]

# pointer for the current working set
ksi = 0

```

```
# Resource Cost
```

```
RCost = 0
```

```
def fuse(n,m):
```

```
    weighted_choices = [('Recycle', (n-1)*(m-1)), ('Success', n+m-2), ('Failure', 1)]
```

```
    population = [val for val, cnt in weighted_choices for i in range(cnt)]
```

```
    return random.choice(population)
```

```
def placeNRec(size):
```

```
    Nksi = int(math.ceil( math.log(size-2,2) ))
```

```
    if WSet[Nksi][0]>0 and WSet[Nksi][1]>0:
```

```
        #print 'ERRROOOORR'
```

```
        sys.exit(1)
```

```
    elif WSet[Nksi][0]>0 and WSet[Nksi][1]==0:
```

```
        WSet[Nksi][1]=size
```

```
    elif WSet[Nksi][0]==0:
```

```
        WSet[Nksi][0]=size
```

```
    else:
```

```
        #print "ERRRROOOORRR --- 2 ---"
```

```
        sys.exit(1)
```

```
    #print "FROM REC", WSet
```

```
def placeNSuc(size):
```

```
    global ksi
```

```
    ksi = ksi+1
```

```
    if WSet[ksi][0]>0 and WSet[ksi][1]>0:
```

```

#print "ERRRROOOORRR --- 3.1 ---"
sys.exit(1)
#print "ERRRROOOORRR --- 3.2 ---"
elif WSet[ksi][0]>0 and WSet[ksi][1]==0:
    WSet[ksi][1]=size
elif WSet[ksi][0]==0:
    WSet[ksi][0]=size
else:
    #print "ERRRROOOORRR --- 4 ---"
    sys.exit(1)
#print "FROM SUC", WSet

def step2():
    global ksi,RCost
    if ksi==0:
        if WSet[ksi][0]==0 and WSet[ksi][1]==0:
            WSet[ksi][0]=3
            #print WSet
            RCost=RCost + 1
            step2()
        elif WSet[ksi][0]!=0 and WSet[ksi][1]==0:
            WSet[ksi][1]=3
            RCost=RCost + 1
            #print WSet
            # print "b"
            step2()
        elif WSet[ksi][0]!=0 and WSet[ksi][1]!=0:
            #print "calling STEP3 from 1"
            step3()
    elif ksi>0:

```

```

if WSet[ksi][0]==0 or WSet[ksi][1]==0:
    ksi = ksi-1
    #print "d"
    step2()
else:
    #print "calling STEP3 from 2"
    step3()

```

```

def step3():
    global ksi, uplimit, FinalCost, RCost
    #print WSet
    n=WSet[ksi][0]
    m=WSet[ksi][1]
    WSet[ksi][0]=0
    WSet[ksi][1]=0
    #print WSet

    FR = fuse(n,m)
    #print FR
    #print RCost
    if FR=='Failure':
        step2()
    elif FR=='Recycle':
        if n>3: placeNRec(n-1)
        if m>3: placeNRec(m-1)
        step2()
    elif FR=='Success':
        if ksi<uplimit-1:
            placeNSuc(n+m-2)
            step2()

```

```

else:
    placeNSuc(n+m-2)
    FinalCost = RCost
    #print WSet
    #print "Reached Set: ", ksi
    #print "Final Log Cost ", math.log10(FinalCost)
else:
    pass
    #print "Bir ihtimal daha var o da olmek mi dersin"

```

```

for i in range(runtimes+1):
    ksi=0
    WSet = [ [0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0],[0,0] ]
    step2()

print (i, "kez calisti")
print ("Total Final Cost=", FinalCost)
print ("Average Cost=", FinalCost/i)
print (math.log10(FinalCost/i))
print (WSet)

```

### 3. Wolfram Mathematica:

(\* 2014-12-22, ALL COMPARISONS,  
 NJP (basic fusion) vs PRA (enhanced fusion) comparison,  
 strategy: similar sizes with RECYCLING,  
 \*)

```

*****NJP("basic          fusion          gate"Oct2011)"*****)
RCostNJPSet1=Log10[1.];SizeNJPSet1=3000/(1000.);"
RCostNJPSet2=Log10[4324/(1000.);SizeNJPSet2=4000/(1000.);"
RCostNJPSet3=Log10[20632/(1000.);SizeNJPSet3=6000/(1000.);"
RCostNJPSet4=Log10[77394/(1000.);SizeNJPSet4=9110/(1000.);"
RCostNJPSet5=Log10[262114/(1000.);SizeNJPSet5=14632/(1000.);"

```

```

RCostNJPSet6=Log10[894658/(1000.);SizeNJPSet6=24688/(1000.);"
RCostNJPSet7=Log10[3069356/(1000.);SizeNJPSet7=44026/(1000.);"
RCostNJPSet8=Log10[11516638/(1000.);SizeNJPSet8=82262/(1000.);"
RCostNJPSet9=Log10[45395526/(1000.);SizeNJPSet9=157810/(1000.);"
NJPList={{SizeNJPSet1,RCostNJPSet1},{SizeNJPSet2,RCostNJPSet2},{SizeNJ
PSet3,RCostNJPSet3},{SizeNJPSet4,RCostNJPSet4},{SizeNJPSet5,RCostNJPSe
t5},{SizeNJPSet6,RCostNJPSet6},{SizeNJPSet7,RCostNJPSet7},{SizeNJPSet8,
RCostNJPSet8},{SizeNJPSet9,RCostNJPSet9}};"
PlotNJPLine=ListLinePlot[NJPList,AxesOrigin→{0,0},PlotRange→{{0,170},{0,
8}}," " PlotStyle→{Black,Opacity[0.3],Dashed}];"
PlotNJPPoint=ListPlot[NJPList,AxesOrigin→{0,0},"
PlotStyle→{Black},PlotMarkers→{"*",16}];"
(*"*****PRAOQuL,"Enhanced fusion",March2013"*****)
RCostPRASet1=Log10[1.];SizePRASet1=3000/(1000.);"
RCostPRASet2=Log10[3632/(1000.);SizePRASet2=4000/(1000.);"
RCostPRASet3=Log10[13554/(1000.);SizePRASet3=6000/(1000.);"
RCostPRASet4=Log10[45408/(1000.);SizePRASet4=9104/(1000.);"
RCostPRASet5=Log10[141194/(1000.);SizePRASet5=14608/(1000.);"
RCostPRASet6=Log10[461412/(1000.);SizePRASet6=24604/(1000.);"
RCostPRASet7=Log10[1529516/(1000.);SizePRASet7=44060/(1000.);"
RCostPRASet8=Log10[5636080/(1000.);SizePRASet8=81970/(1000.);"
RCostPRASet9=Log10[21452106/(1000.);SizePRASet9=158246/(1000.);"
PRAList={{SizePRASet1,RCostPRASet1},{SizePRASet2,RCostPRASet2},{Size
PRASet3,RCostPRASet3},{SizePRASet4,RCostPRASet4},{SizePRASet5,RCost
PRASet5},{SizePRASet6,RCostPRASet6},{SizePRASet7,RCostPRASet7},{Size
PRASet8,RCostPRASet8},{SizePRASet9,RCostPRASet9}};"
PlotPRALine=ListLinePlot[PRAList,AxesOrigin→{0,0},PlotRange→{{0,150},{
0,8}}," " PlotStyle→{RGBColor[0,1,0],Opacity[0.3],Dashed}];"
PlotPRAPoint=ListPlot[PRAList,AxesOrigin→{0,0},"
PlotStyle→{RGBColor[0,1,0]},PlotMarkers→{"*",16}];"
(*"*****QIPOQuL,"4-fusion",Sept2013"*****)
RCostQIPSet1=Log10[1];SizeQIPSet1=3000/(1000.);"
RCostQIPSet2=Log10[39.548];SizeQIPSet2=8000/(1000.);"
RCostQIPSet3=Log10[6505.492];SizeQIPSet3=21696/(1000.);"
RCostQIPSet4=Log10[1.1360491398× [[10] ^7];SizeQIPSet4=72123/(1000.);"
QIPList={{SizeQIPSet1,RCostQIPSet1},{SizeQIPSet2,RCostQIPSet2},{SizeQIP
Set3,RCostQIPSet3},{SizeQIPSet4,RCostQIPSet4}};"
PlotQIPLine=ListLinePlot[QIPList,AxesOrigin→{0,0},PlotRange→{{0,150},{0,
8}}," " PlotStyle→{RGBColor[1,0,0],Opacity[0.3],Dashed}];"
PlotQIPPoint=ListPlot[QIPList,AxesOrigin→{0,0},"
PlotStyle→{RGBColor[1,0,0],AbsolutePointSize[10]},PlotMarkers→{"*",16}];"
(*"*****BoostOQuL&Sahin-San&Mark,"3-fusion",PresentWork"*****)
RCostBoostSet1=Log10[1];SizeBoostSet1=3000/(1000.);"
RCostBoostSet2=Log10[5.667];SizeBoostSet2=4842/(1000.);"
RCostBoostSet3=Log10[32.753];SizeBoostSet3=9274/(1000.);"
RCostBoostSet4=Log10[237.854];SizeBoostSet4=19802/(1000.);"
RCostBoostSet5=Log10[2480.618];SizeBoostSet5=51054/(1000.);"
RCostBoostSet6=Log10[34290.857];SizeBoostSet6=142734/(1000.);"
BoostList={{SizeBoostSet1,RCostBoostSet1},{SizeBoostSet2,RCostBoostSet2},

```

```

{SizeBoostSet3,RCostBoostSet3},{SizeBoostSet4,RCostBoostSet4},{SizeBoostS
et5,RCostBoostSet5},{SizeBoostSet6,RCostBoostSet6}};"
PlotBoostLine=ListLinePlot[BoostList,AxesOrigin→{0,0},"
PlotStyle→{RGBColor[0,0,1],Opacity[0.3]}];"
PlotBoostPoint=ListPlot[BoostList,AxesOrigin→{0,0},"
PlotStyle→{RGBColor[0,0,1]},PlotMarkers→{"+",16}];"
Show[PlotNJPLine,PlotNJPPoint,PlotPRALine,PlotPRAPoint,PlotQIPLine,PlotQ
IPPoint,PlotBoostLine,PlotBoostPoint," " AxesLabel→{"W State: \n Size:
N","Resource:\n"log_10 "R["W_N "]}," " AxesStyle→Directive[Black,12]]

```



## ÖZGEÇMİŞ



### Kişisel Bilgiler

Adı Soyadı	Sinan Buğu
Uyruğu	T.C
Doğum yılı, Yeri	1986,Hazro
Telefon	05062086899
E-mail	sinanbugu@gmail.com

### Eğitim

Derece	Kurum/Anabilim Dalı Programı	Yılı
Yüksek Lisans	İ.Ü Fen Bilimleri Enstitüsü/ Bilgisayar Mühendisliği Anabilim Dalı/Bilgisayar Mühendisliği Programı	2015
Lisans	İ.Ü Fen Fakültesi / Matematik Bölümü	2011
Lise	M. Adnan Özçelik Anadolu Lisesi	2004

### Makaleler/Bildiriler

Tez Kapsamında yapılan SCI-E dergi yayınları

Bugu S., Yesilyurt C., Ozaydin F., “Enhancing the W-state Quantum-Network-Fusion Process with a Single Fredkin Gate”. *Physical Review A*, 87, 032331, (2013),

Yesilyurt C., Bugu S., Ozaydin F., “An Optical Gate for Simultaneous Fusion of Four Photonic W or Bell States”, *Quantum Information Processing*, 12(9), 2965 (2013).

Ozaydin F., Bugu S., Yesilyurt C., Altintas A. A., Tame M., Özdemir Ş. K., “Fusing multiple W states simultaneously with a Fredkin gate”, *Physical Review A* 89, 042311 (2014)

Tez kapsamında yapılan uluslararası konferans bilgileri

Bugu S., Yesilyurt C., Altintas A. A., Ozaydin F., "Fusing Several Polarization Based Entangled Photonic W State" *QIPC (Quantum Information Processing and Communication Conference)* 30 June - 5 July 2013 Florence, Italy

Bugu S., Erol V., Altintas A. A., Ozaydin F. , "Strategy With Recycling For The Enhanced Setup for Creating Large-Scale W State Networks" *QCN (Quantum Communication Networks Workshop)* 9-10 January 2014, Leeds, United Kingdom.

Yesilyurt C., Bugu S., Altintas A. A., Ozaydin F., "Creation of Four Photon W states, out of Two Bell Pairs" *Apmas 2014* 24-27 April 2014, , Muğla, Turkey

Tez dışındaki çalışmalardan yapılan SCI-E dergi yayınları

Ozaydin F., Altintas A. A., Bugu S., Yesilyurt C. "Quantum Fisher Information of N Particles in the Superposition of W and GHZ States". *International Journal of Theoretical Physics*, 52, 2977 (2013),

Altintas A. A., Ozaydin F., Yesilyurt C. Bugu S., Arik M. "Constructing quantum logic gates using q-deformed harmonic oscillator algebras", *Quantum Information Processing*, 13 (4), 1035 (2014).

Ozaydin F., Altintas A. A., Bugu S., Yesilyurt, C. , Arik M. "Quantum Fisher Information of Several Qubits of a GHZ and two W States with Arbitrary Relative Phase". *International Journal of Theoretical Physics*, (2014) DOI: 10.1007/s10773-014-2119-4

Ozaydin F., Altintas A. A., Bugu S., Yesilyurt C. "Behavior of Quantum Fisher Information of Bell Pairs under Decoherence Channels". (Proceedings of APMAS2014 International Conference paper selected and published in A. Phys. Pol. A.)

Tez dışındaki çalışmalardan yapılan uluslararası konferans bilgileri

Ozaydin F., Altintas A. A., Yesilyurt C. , Bugu S., "Quantum Fisher Information of Bipartitions of W States" 24-27 April 2014, *Apmas 2014* , Muğla, Turkey

Erol V., Bugu S., Ozaydin F. , Altintas A. A., "An Analysis of Concurrence Entanglement Measure and Quantum Fisher Information of Quantum Communication Networks of Two-Qubits" 23-25 April 2014, *IEEE - Signal Processing Applications* (SIU in Turkish) Trabzon, Turkey