Bir Steganografi Sisteminin FPGA Üzerinde Gerçeklenmesi

Betül ELÇİ, Berna ÖRS, Volkan DALMIŞLI

***Özet*— Bu çalışmada, öncelikle steganografinin içeriği ve uygulama alanları incelenmiştir. İlk olarak bir kölenin saçının kazıtılarak, bilginin dövme şeklinde kölenin kafa derisine işlenmesi ve kölenin saçı yeteri kadar uzadığında bilginin tekrar elde edilmesi ile başlayan, veri gizleme bilimi olarak adlandırılan steganografi, günümüzde teknolojik gelişmelerle çok ileri boyutlara ulaşmıştır. Veriyi gizlerken ve veriyi tekrar elde ederken kullanılan yöntemler her geçen gün gelişme göstermektedir. Böylece verinin en güvenli şekilde iletimi sağlanmaya çalışılmaktadır.**

Bu çalışmada incelenen üç steganografi yöntemi uygulanmıştır. Bu yöntemleri gerçeklerken VHDL donanım tanımlama dili kullnılmıştır. Steganografi sisteminin FPGA üzerinde tasarımı ve gerçeklenmesi sağlanmıştır. Gerçeklenen üç yöntemin sonuçları kıyaslanarak avantajları ve dezavantajları yorumlanmıştır.

Anahtar Kelimeler—Steganografi, FPGA, VHDL

I. GİRİŞ

Bu çalışmada, steganografi sisteminin FPGA üzerinde tasarımı ve gerçeklenmesi sağlanmıştır. Eski Yunancada gizlenmiş yazı anlamına gelen steganografi, bilginin görünürlüğünü gizleme bilimine verilen isimdir [1]. Günümüzde karşılaşılan en büyük yanlış anlama steganografinin şifreleme ile karıştırılmasıdır. Veriyi gizleme sanatı olarak bilinen bu bilimin şifrelemeye göre en büyük üstünlüğü bilgiyi gören bir kimsenin gördüğü şeyin içinde önemli bir bilgi olduğunu fark edemiyor olmasıdır, böylece içinde bir bilgi aramaz oysa bir şifreli mesaj, çözmesi zor olsa bile, gizemi dolayısıyla ilgi çeker. Çünkü bir bilginin gizlendiği bellidir [3]. Günümüzde steganografi bilimi sayesinde ses, video, resim dosyalarına, disketlere ve haberleşme kanallarına istenilen veri gizlenebilmektedir [1].

Çalışmada steganografinin resim alanındaki uygulamaları donanımsal olarak gerçeklenmiştir.

Steganografide resmin içine veri gizlerken en çok kullanılan yöntem resmi oluşturan piksellerin en anlamsız bitlerinin kullanıldığı uygulamalardır. Bunun nedeni en düşük anlamlı bitte yapılan değişiklikleri insan gözünün fark edememesidir. Yani en anlamsız bitlere saklanan veriyi içeren resimle, esas resim yan yana konulduğunda gözle görülür hiçbir fark yoktur. Bu durumdan yola çıkılarak steganografide resmin içine veri gizlenirken çeşitli yöntemler geliştirilmiştir [1].

Bu çalışmada bu yöntemlerden üç tanesi kullanılmıştır. Bunlardan birincisinde bir resmi oluşturan tüm piksellerin son bitleri değiştirilmiş ve bu değişimin bir fark yaratmadığı anlaşılmış, ikincisinde ilk pikselden başlanarak veri gizlenmiş daha sonra da aynı şekilde çözümlenmesi yapılmıştır. Son yöntemde ise verinin hangi piksellere gizleneceğini belirten bir steganografik anahtar kullanılarak veri gizlenmiş ve tekrar elde edilmiştir.

II. STEGANOGRAFİ

Steganografi veriyi gizlemenin ve zararsız taşıyıcılarla veriyi taşımanın sanatıdır [1]. Kelime anlamı olarak kökleri ve γραΦειν'ye ait olmakta ve Yunan alfabesinden gelmekte olup kaplanmış yazı anlamına gelmektedir. Yunanca steganos sözcüğü gizli, saklı ve grafi sözcüğü çizim ya da yazım demektir. Bu sanat var olan veriyi gizlemede birçok gizli haberleşme tekniği kullanmaktadır. Steganografi eski bir el sanatı olmasına rağmen günümüzde bilgisayar teknolojisiyle yeni bir içerik kazanmıştır. Bilgisayar tabanlı steganografi teknikleriyle veriyi yeni gizleme teknikleri geliştirilmiştir.

Bilgiler metin formunda, sayısal 1 ve 0 olarak ya da başka çeşitlerde iletime geçerken verinin kime ait olduğunu belirten bir çeşit parmak izi bırakırlar. Steganografi bir çeşit kriptografi yötemi gibi düşünülebilir. Her ikisi de haberleşme sırasında kaydedilmiş veriye bilgi ekleyerek çalışırlar. Kriptografi teknikleri bilgiyi belirli algoritmalara dayanarak şifreleyip güvenli bir örtü yaratmayı amaçlar. Steganografi ise kriptografiden farklı olarak veriyi örterek gizlemeyi sağlar. kriptografide şifreli metin olarak adlandırdığımız örtülü yapı dikkat çekebilirken steganografide kendini gizlediğinden dikkat çekmemeyi sağlar. Bu da verinin güvenli bir şekilde taşınması açısından önemli ve yararlı bir durum oluşturmaktadır [7].

III. STEGANOGRAFİ UYGULAMASI

Günümüzde resim alanında yapılan steganografi uygulamalarında çeşitli algoritmalar ve yöntemler kullanılmaktadır. Bunlardan en yaygın olanı resmi oluşturan piksellerin en düşük anlamlı bitinde yapılan değişiklikleri kapsar. Bu çalışmam kapsamında bu yöntemlerden üç tanesi yer almaktadır.

B. Elçi İstanbul Teknik Üniversitesi öğrencisidir.

B. Örs Istanbul Teknik Universitesi, Elektrik-Elektronik Fakultesi, Elektronik ve Haberlesme Muhendisligi Bolumu, 34469 Maslak, Istanbul'da Yardımcı Doçent Dr. olarak görev yapmaktadır. Tel: 0212 285 36 03, Fax: 0212 285 35 65, url: www2.itu.edu.tr/~orssi.

V. Dalmışlı İstanbul Teknik Üniversitesi öğrencisidir.

A. Uygulamada Kullanılan Yöntemler

Çalışmada resmi oluşturulan piksellerin son bitinde yapılan değişikliklerle üç yöntem gerçeklenmiştir. Gerçeklenme aşamasından sonra elde edilen resimlerle esas resimlerle kıyaslandığında gerçekten gözle görülür bir fark olmadığı ortaya çıkmıştır. Çalışma esnasında en yüksek verim alınabilmesi için resimler için kayıpsız bir sıkıştırma algoritması olan bitmap formatı seçilmiştir.

1) En Düşük Anlamlı Bitlerin Tümünün Değiştirilmesi Yöntemi

Bu yöntemle steganografinin kapsamında anlatılanları gerçekleyerek gerçekten kullanılan ve değişen resimler arasında fark olmadığını görülmesi sağlanmıştır. Öncelikle resimler piksellerden oluştuğu ve matris halinde bu pikseller dizildiği için bir metin dosyasına aktarım işlemi yapılması gerekiyordu. Bunun için wnbrowse editörü ile Şekil 1'deki resmin hex formatında görülmesi sağlanmıştır.

Şekil 1 F15 Uçak Resmi [4]

Oluşan 205×138'lik resim içinde sadece hex formatında pikseller içeren bir yapı oluşmuştur. Şekil 2'de bu matrisin sadece ilk 10 x 10'luk kısmı alınmıştır.

Burada 8 bitlik yapılardan yani baytlardan oluşmuş bir matris görülmektedir. Bu matriste her satır ve sütun 10 bayttan oluşmaktadır.

8 bitlik bu yapıya resimde piksel adı verilmektedir. Resmin boyutu değiştikçe matris boyutu da dolayısıyla resmi oluşturan piksel sayısı da değişmektedir [2].

42	4D	76	30	05	00	00	00	00	00
00	00	9A	01	00	00	14	01	00	00
00	00	00	00	00	00	C4	0E	00	00
00	00	00	00	00	00	C0	86	48	C4
86	49	C1	84	46	C2	85	49	C1	85
47	C1	86	4B	C1	84	46	C4	84	46
C1	86	46	C3	86	45	C0	86	48	C2
89	44	C4	87	43	C3	86	45	C3	85
49	C4	87	45	C2	88	47	C5	8A	49
C2	86	4A	C4	84	46	C2	87	47	C0
82	45	BF	81	46	BD	85	47	C2	87

Şekil 2 Orijinal resmin ilk 10×10'lık kısmının heksadesimal formatında Görünümü

Steganografideki en yaygın yöntem olan en düşük anlamlı bitin değiştirilerek uygulanmasını burada matriste gerçeklenmesi sağlanmıştır. Yöntemde belirtilen yapıya göre bir resmi oluşturan piksellerin her birinin son bitlerini değiştirilirse yani 0 ise 1, 1 ise 0 yapılırsa yeni oluşan resimle eskisi kıyaslanırsa arada gözle görülür bir fakın olmayacağıdır [7].

Şekil 2'deki kaynak resmin ilk 10×10'luk kısmının piksellerine bakacak olursak her pikselin son biti 1 ise 0, 0 ise 1 yapılarak yöntem gerçeklenmiştir. Bunun için her hex yapısı ikilik düzende düşünülmüştür. Bu işlemin resimdeki her piksele uygulanması sağlanmıştır ve Şekil 3'deki matris elde edilmiştir.

43	4C	77	31	04	01	01	01	01	01
01	01	9B	00	01	01	15	00	01	01
01	01	01	01	01	01	C5	0F	01	01
01	01	01	01	01	01	C1	87	49	C5
87	48	C0	85	47	C3	84	48	C0	84
46	C0	87	4A	C0	85	47	C5	85	47
C0	87	47	C2	87	44	C1	87	49	C3
88	45	C5	86	42	C2	87	44	C2	84
48	C5	86	44	C3	89	46	C4	8B	48
C3	87	4B	C5	85	47	C3	86	46	C1
83	44	BE	80	47	BC	84	46	C3	86

Şekil 3 Son Bitleri Değişmiş Resmin İlk 10×10'luk Kısmının Görünümü

Sadece son bit değiştirilmesi için kombinezonsal bir devre tasarımı yapılmıştır. Bundan sonra gereken işlem 205×138 formatındaki piksellerden resim elde etmektir. Ancak bu şekilde bitleri değiştirilmeden önceki resimle farkının olup olmadığı karşılaştırabilmektedir [2].

Bu aşamada tasarlanan devrenin benzetimi devreye Şekil 1'deki resmin piksel değerleri giriş olarak verilerek yapılmıştır. Sonuçta tüm piksellerin son bitleri değişmiş haliyle elde edilen resim Şekil 4'deki gibidir.

Şekil 4 F15 Uçak Resmi

Bu iki resimde burada da görüldüğü gibi gözle görülür hiçbir fark yoktur.

2) Bir Veriyi En Düşük Anlamlı Bitlere Gömme ve Elde Etme Yöntemi

Bir resmi oluşturan tüm piksellerin son bitlerinin değişimi resimde fark edilir bir ayrım yaratmadığından ilerleyen steganografik çalışmalarda bu en düşük anlamlı bitlerle veri gömmeye başlanmıştır. Bu konuda çeşitli algoritmalar geliştirilmiştir [3]. Bu çalışmada resmin içine veri gizlenmiş ve daha sonra bu resim iletilmiştir. Resmi alan kullanıcı resmin içine gömüldüğü gibi veriyi çıkartmış ve böylece haberleşme sağlanmıştır. Bu yöntemde veriyi gömme işlemi Algoritma 1'de gösterildiği gibidir.

Bu yöntemde önceki yöntem gibi resmin wnbrowse editörü gibi bir araç ile hex formatında gösterimi sağlanır ve bir metin

dosyasına kaydedilir. Elde edilen bu esas resim *cover* olarak ifade edilir. Esas resmin baytlar c_i olarak ifade edilmektedir. Steganografik tanım olarak steganografik anahtar ifadesi s_i olarak belirtilmiştir. Resmin içine gömülmek istenen verinin bitlerini ifade etmek için m_i kullanılmıştır. Bu algoritmaya göre veri gizlenirken resmi oluşturan ilk baytın son bitinde değişiklik yapılarak veri gömülmeye başlanır. Yani veriyi gömme işlemi ilk baytın en düşük anlamlı bitinden başlanarak sırayla hangi baytın en düşük anlamlı bitine kadar gidiyorsa yapılır. Bu çalışmada, veri gizlenirken alfabedeki harflerin ikilik düzende karşılıkları şeklinde bir dönüşüm kullanılması gerekmektedir. Bunu sağlamak için ascii karakter kodlamasından yararlanılmıştır [2]. Karşı taraf veriyi elde ettiğinde ikilik düzenin karşılığını alfabede bulunan 29 harfin karşılığı olarak elde etmesi içinse Matlab programında yazılan kod parçası kullanılmıştır. Örnek bir uygulama olarak olarak *itu* sözcüğünü gizlemeye çalıştığımızda ilk olarak yapılan işlem *itu* kelimesinin ascii karakter kodlamasında karşılığını bulmak olacaktır.

Algoritma 1

Giriş: $c = (c_1 c_2 \cdots c_n)_8,\ m = (m_1 m_2 \cdots m_l)_2$
Çıkış: $s = (s_1 s_2 \cdots s_n)_8$
for $i = 1, \cdots, l$
  $s_i \leftarrow c_i$
  $s_{i8} \leftarrow m_i$
for $i = l+1, \cdots, n$
  $s_i \leftarrow c_i$

itu kelimesinin ascii karakter kodlamasındaki karşılığı bulunur. Buna göre i = 69, t = 74, u = 75 şeklinde belirlenir. Burada 69, 74 ve 75 şeklinde gördüğümüz sayılar hex formatında yapılardır. Veriyi resmin son bitlerine gizlemek için ikilik düzendeki ifadesine ihtiyacımız var. Bu nedenle *itu* sözcüğünün ikilik düzendeki karşılıkları i = 0110 1001, t = 0111 0100, u = 0111 0101 şeklinde belirlenir. Bundan sonraki işlemi bu bitleri verinin gizleneceği resme ilk baytın son bitlerinden başlayarak sırayla gömmektir. Alfabede bulunan üç harfi gizlemek için her bir harfin 8 bitlik karşılığı olduğundan 24 bite ihtiyaç vardır. Bu da resmimizin ilk baytından başlanarak sırayla 24 baytın son bitlerine veriyi gizlemek anlamına gelmektedir. F15 uçak resminin ilk 10×10'luk kısmındaki ilk bayt, 42 idi. Yani 0100 0010 şeklinde ikilik düzende ifadesi vardır. Gizlemek istediğimiz "itu" sözcüğünün ise ilk karakteri i = 0110 1001 şeklinde ifade edilmişti. Demek ki ilk baytın son bitine yani 42 in son bitine *i* karakterinin ilk biti gömülmelidir. 42 = 0100 001*0* olduğundan son bit 0'dır. i = *0*110 1001 olduğundan ilk bit 0'dır. Demek ki kaynak resmin ilk baytında bir değişiklik olmayacaktır. Yine 0 olarak kalacaktır. Bu şekilde bu durum sırayla devam edecektir. Daha sonraki aşama *i* karakterinin ikinci biti olan 1'i ana resmin ikinci baytının son bitine gömmek olacaktır. Gömmek istenilen 24 bit resmin ilk baytından başlanarak 24 baytın en düşük anlamlı bitlerine yerleştirilecektir.

Burada algoritmada da belirtildiği gibi resmin bitlerini kontrol eden bir yapı yoktur. Yani *i* karakterinin ilk biti 0, aynı zamanda resmin ilk baytının da son biti 0 diyen bir kontrol mekanizmasına gerek yoktur. Saklamak istediğimiz bilgi ne ise onu oluşturan bitleri teker teker piksellerin son bitlerine yazdırmak yeterlidir [2].

Veriyi gömme işlemi bittikten sonra tekrar Matlab programı kullanarak bu matrisi bitmap formatında resme dönüştürürüz. Yeni oluşan resimle eskisi kıyaslandığında gözle görülür bir fark olmamaktadır.

Bu içinde veri gömülü resim iletilmek istenen kişiye iletildiğinde o kişinin bu bilgiye ulaşması için tekrar o bilgiyi elde etmesi gerekir. Bunun için kullanılan algoritma aşağıdaki gibidir.

Algoritma 2

Giriş: $s = (s_1 s_2 \cdots s_n)_8$
Çıkış: $m = (m_1 m_2 \cdots m_l)_2$
for $i = 1, \cdots, l$
  $m_i \leftarrow s_{i8}$

İçine veri gizlenmiş olan resim iletilmek istenen kişinin eline geçtiğinde veriyi gönderenden gerekli bilgileri önceden aldığından, veriye ulaşmak için Algoritma 2'nin kullanılması yeterlidir. Bu gerekli bilgi verinin alfabetik olarak ya da bit sayısı olarak uzunluğunu içermektedir.

Veri uzunluğunu bilen kullanıcı, verinin içerdiği bit sayısı kadar matrisi oluşturan baytlardan baştan başlayarak alır. Aldığı bu baytların son bitlerini bir metin dosyasına çıkartır ve ascii karakter kodlamasına göre her baytın karşılığını alfabedeki harfler olarak elde etmelidir [2]. Bunun için de Matlab programında her harfin ascii karakter kodlamasını içeren bir kod parçası kullanarak bunun elde edilmesi sağlanır.

Günümüzde steganografinin çok fazla gelişme imkânı bulmasıyla buna karşı geliştirilen atak türlerinde de gelişme olmuştur. Bu nedenle böyle bir veri örneğin internet ortamında dolaşırken çok da fazla güvenli olmaz. Çünkü resmin içinde veri var mı diye bakan analiz yapmak isteyen kişi ilk baytın son bitlerinden başlayarak bir çözümleme yapmaya çalışabilir. Bu da verinin güvenliğini önemli ölçüde tehdit eden bir yapı oluşturur. Bu nedenle steganografinin resim alanındaki birçok çalışmasında bu yöntem kullanılsa da saldırı ve analiz tekniklerinin de geliştirilmesiyle veriyi gömme - çıkarma alanında yeni algoritmalar geliştirilmiştir [2,3].

Şekil 5 Veri gömülmeden önce

Şekil 6 Veri gömüldükten sonra

Görüldüğü gibi *itu* sözcüğü gömülmeden önce Şekil 6'daki resimle, gömüldükten sonraki halini yani Şekil 7'deki resmi yan yana koyduğumuzda gözle görülemeyecek kadar değişim oluştuğundan fark edilememektedir.

3) Rastgele Aralık Yöntemi

Bir önceki yöntemin zayıf yanı iletilmek istenilen verinin başkaları tarafından ele geçirilme durumunun olmasıydı. Bu ihtimali azaltmak için geliştirilen algoritmalardan birisi "Rastgele Aralık Yöntemi" dir. Bu yöntemin bir önceki yöntemden farkı gizlenecek verinin bitlerinin ilk bayttan başlanarak sırayla en düşük anlamlı bitlere yüklenmeyecek olmasındandır. Rastgele Aralık Yöntemi'nde hangi piksellere verinin gizleyeceği belirli denklemlere bağlıdır [2]. Bu yöntemde kullanılan algoritma aşağıdaki gibidir.

Algoritma 3

Giriş: $c=(c_1c_2\cdots c_n)_8$, $m=(m_1m_2\cdots m_l)_2$

Çıkış: $s=(s_1s_2\cdots s_n)_8$

for $i=1,\cdots,n$

$s_i \leftarrow c_i$

Rastgele $1\le k_i\le n$ olmak üzere l adet k değeri belirlenir.

$j_1=k_1$

for $i=1,\cdots,l$

$j_i=j_{i-1}+k_i \bmod n$

for $i=j_1,\cdots,j_l$

$s_{i8} \leftarrow m_i$

Bu yöntemin en önemli özelliği veri gizlenirken bir steganografik anahtarın kullanılmasıdır. Steganografik anahtar verinin rastgele olarak hangi piksellerin en düşük anlamlı bitlerine gizleneceğini belirtir. Burada önemli olan hem veriyi gönderen hem veriyi alan kişi için steganografik anahtardır. İki taraf da aynı steganografik anahtar olmadan veri iletimini sağlayamazlar [2].

Çalışma sırasında steganografik anahtar belirleme de çeşitli yöntemler araştırılmıştır. Bu yöntem dahilinde Matlab programı ile rastgele sayılar üretilmiştir. Bu sayıların steganografik anahtar olmasına karar verilmiştir. Matlab programı kullanılarak bu anahtarı üretilme sebebi bunun gerçekten diğer algoritmalardan daha hızlı ve kolay bir yol olmasıdır.

Rastgele aralık yöntemine dayalı olarak steganografik anahtarın uzunluğu gizlenmek istenen verinin uzunluğu ile aynı olmalıdır. Bu yöntemde veriyi gömme işlemine bir önceki yöntemdeki gibi başlanır [2].

Burada gizlenmek istenen verinin bitleri yine resmin piksellerinin en son bitlerine gömülecektir. Yalnız bu yöntemde ilk bayttan başlanarak sırayla gömme işlemi yapmak yerine hangi baytlara verinin gizleneceğini gösteren bir steganografik anahtarın kullanıldığı denklem yer almaktadır [2]. Burada saklamak istediğimiz bilgi toplam 24 bitten oluşmaktadır. Bu da oluşturmamız gereken anahtarın 24 bit uzunluğunda olması gerektiğini belirtir. Bu steganografik anahtar belirlendikten sonra yönteme dayalı olarak veriyi resmin hangi baytlarına saklanıldığını belirtildiği denklem 4 kullanılır.

$$j_1=k_1 \qquad (4)$$
$$j_i=j_{i-1}+k_i \bmod n$$

Bu denklemde j_i rastgele belirlenmiş steganografik anahtarı buna bağlı olarak 4 denklemini kullanarak resmin hangi baytlarına verinin gizleneceğini belirten bir indistir [2].

Örneğin Matlab programını kullanarak *itu* sözcüğün *i* harfini gizlemek için 8 tane rastgele sayı yani 4 denklemindeki k değerlerini belirlensin. Bu sayılar 1, 3, 12, 56, 9, 10, 18, 22 şeklinde belirlendi. Buradan anlaşıldığı gibi k değerleri sırasıyla rastgele belirlenen 8 değerden oluşmaktadır. Belirlenen bu k değerlerinden yararlanarak verinin gizleneceği j değerleri 4 denklemine göre belirlenir.

$k_1=1 \rightarrow j_1=1$

$k_2=3 \rightarrow j_2=j_1+k_2=4$

$k_3=12 \rightarrow j_3=j_2+k_3=16$

$k_4=56 \rightarrow j_4=j_3+k_4=72$

$k_5=9 \rightarrow j_5=j_4+k_5=81$

$k_6=10 \rightarrow j_6=j_5+k_6=91$

$k_7=18 \rightarrow j_7=j_6+k_7=109$

$k_8=22 \rightarrow j_8=j_7+k_8=131$

Belirlenen k dolayısıyla j değerleri resmin bayt numaralarını göstermiş oldu. Buradan çıkarılması gereken sonuç *i* harfini gizleme işleminin, resmin 1, 4, 16, 72, 81, 91, 109 ve 131 numaralı baytlarında gerçekleşmesi gerektiğidir. Numarası belirtilmiş baytların en düşük anlamlı bitlerine sırasıyla 0110 1001 bitleri gömülür. Aynı işlemler diğer karakterler için de steganografik anahtar ve verilen denklem yardımıyla hangi baytlara gömme işleminin yapılacağı belirlenerek uygulanır. Gömme işlemi rastgele aralık yöntemine göre gerçeklenmiştir. İçine veri gömülen resim iletilmek istenen yere ulaştığında bu veriyi tekrar elde etmek için de bu yöntem dahilinde bir algoritma vardır.

Burada unutulmamalıdır ki hem veriyi gönderen hem veriyi alan kişilerde aynı steganografik anahtar bulunmaktadır [2].

Steganografik anahtarın her iki tarafta da olduğu kabul edilen bu yöntem gereği veriyi resimden çıkartma algoritması aşağıdaki gibidir.

Algoritma 4

Giriş: $s = (s_1 s_2 \cdots s_n)_8,\ k_1, k_2, \cdots, k_l$

Çıkış: $m = (m_1 m_2 \cdots m_l)_2$

$j_1 = k_1$

for $i = 1, \cdots, l$

$j_i = j_{i-1} + k_i \bmod n$

for $i = j_1, \cdots, j_l$

$m_i \leftarrow s_{i8}$

Algoritma 4'de ifade edildiği gibi gömülmüş veriyi tekrar elde ederken öncelikle hangi baytlara verinin gizlenmiş olduğunun bulunması gerekir. Bu nedenle elinde steganografik anahtar değerleri olan kişi denklem 4 gereği j_i değerlerini yani hangi baytlara veri gizlendiğini belirler.

Şekil 7 Veri gizlenmeden önce

Şekil 8 Veri gizlendikten sonra

Daha sonra belirlenen baytların en düşük anlamlı bitleri yan yana getirilerek gizlenmiş olan veri elde edilmiş olur [2]. Burada yapılması gereken son işlem elde ettiğimiz veriyi alfabe düzenine geçirmektir. Bunun için de bir önceki yöntem de olduğu gibi Matlab programı yardımıyla gerekli dönüşüm yapılır ve alfabetik karşılıklar elde edilir. Şekil 7 ve Şekil 8'de görüldüğü gibi algoritma gerçeklendikten sonra resmin içine veri gömüldüğünde gözle görülür bir fark olmamıştır.

IV. SONUÇ

İlk yöntem gerçeklendiğinde resmi oluşturan baytların en düşük anlamlı biti değiştirildiğinde ve yeni yapı tekrar resim haline getirildiğinde resimde gözle görülür bir fark olmadığı görülmüştür. Resmi oluşturan piksellerin en anlamsız bitine bir veri gömülüp tekrar elde etme işlemi yapıldı. Yani bir yerden bir yere verinin resim aracılığıyla güvenli bir şekilde taşınabileceği ve verinin tekrar gömüldüğü gibi çıkarılabileceği gösterildi ve FPGA üzerinde gerçeklendi.

Yapılan araştırmalar sonucunda steganografi teknikleri ve bu alanda kullanılan algoritmalar geliştikçe stegoanaliz tekniklerinin de zamanla geliştiği görülmüştür. Bu nedenle bu algoritmanın aslında zayıf bir yanın olduğu belirtilmelidir. Bu zayıf yan veri gömülme işlemi sırasında ilk baytın en düşük anlamlı bitinden başlanarak sırayla verinin gömülme işleminin uygulanmasındandır. Bu nedenle steganografi alanında kullanılan başka yöntemler de araştırılmış ve gerçeklenem için rastgele aralık yöntemi seçilmiştir. Bu yöntemde veri gizlenirken güvenliği sağlamak adına steganografik bir anahtar kullanılır. Kullanılan bu anahtar sayesinde algoritmaya bağlı olarak verinin hangi baytların en anlamsız bitine gizleneceği belirlenir. Bu sayede elinde steganografik anahtar olmayan birinin veriyi ele geçirmesi çok güçtür.

Yukarıda anlatılan her üç steganografi yöntemi VHDL tanımlama dili kullanılarak tasarlandı ve FPGA üzerinde gerçeklendi. Benzetimleri yapılarak sonuçların beklendiği gibi olduğu görüldü.

V. KAYNAKLAR

[1] Johnson, N. F., Duric Z. ve Jajodia S., 2001. Information Hiding : Steganography and Watermarking - Attacks and Countermeasures, Boston.

[2] Katzenbeisser, S. ve Petitcolas, Fabien A. P., 2000. Information Hiding Techniques for Steganography and Digital Watermarking, London.

[3] Marvel, L. M., *Information Hiding: Steganography and Watermarking,* 2005, Optical and Digital Techniques for Information Security

[4] Petitcolas, Fabien A. P., 2008. The Image Downgrading Problem, http://petitcolas.net/fabien/steganography/image_downgrading/index.html.