FPGA İLE VERİ GİZLEME UYGULAMALARI

Zeynep PEKER

T.C.

ULUDAĞ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

FPGA İLE VERİ GİZLEME UYGULAMALARI

Zeynep PEKER

Yrd.Doç.Dr. Ersen YILMAZ

(Danışman)

YÜKSEK LİSANS TEZİ

ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2013

TEZ ONAYI

Zeynep PEKER tarafından hazırlanan " FPGA İLE VERİ GİZLEME UYGULAMALARI " adlı tez çalışması aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Uludağ Üniversitesi Elektronik Mühendisliği Anabilim Dalı' nda **YÜKSEK LİSANS** tezi olarak kabul edilmiştir.

Danışman : Yrd.Doç.Dr. Ersen YILMAZ

Başkan : Yrd.Doç.Dr. Ersen YILMAZ İmza
U.Ü. Müh.- Mim.Fakültesi,
Elektronik Mühendisliği Anabilim Dalı

Üye : Prof.Dr.Erdoğan DİLAVEROĞLU İmza
U.Ü. Müh.- Mim.Fakültesi,
Elektronik Mühendisliği Anabilim Dalı

Üye : Yrd.Doç.Dr. Fatih KARPAT İmza
U.Ü.Müh.- Mim.Fakültesi,
Konstrüksiyon Ve İmalat Anabilim Dalı

Yukarıdaki sonucu onaylarım

Prof.Dr.Kadri ASLAN

Enstitü Müdürü

……/……/……..

U.Ü. Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,

- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,

- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,

- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,

- kullanılan verilerde herhangi bir tahrifat yapmadığımı,

- bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

14/02/2013

Zeynep PEKER

ÖZET

Yüksek Lisans Tezi

FPGA İLE VERİ GİZLEME UYGULAMALARI

Zeynep PEKER

Uludağ Üniversitesi

Fen Bilimleri Enstitüsü

Elektronik Mühendisliği Anabilim Dalı

Danışman : Yrd. Doç. Dr. Ersen YILMAZ

Yüksek lisans tez çalışmasında veri gizleme uygulamalarının FPGA platformu üzerinde gerçekleştirilmesi amaçlanmıştır. FPGA olarak Xilinx firmasına ait Spartan 3E geliştirme kartı seçilmiş ve FPGA geliştirme kartı üzerinde yer alan 2*16 satır LCD ekran kullanılmıştır.

Tez çalışmasında öncelikle FPGA' in tarihler boyunca gelişimi, özellikleri ve avantajlarından bahsedilmiştir. Bununla birlikte donanım tanımlama dillerinden, tez çalışmasında kullanılan donanım tanımlama dili olan VHDL'in özellikleri ve avantajlarına detaylı biçimde yer verilmiştir. FPGA programlama temelleri ve Xilinx firması tarafından kullanıcılara sunulan ISE Design Suite 12.1 programı anlatılmıştır. Sonrasında veri gizleme çalışmalarının tarihi, alt dalları ve steganografinin öneminden bahsedilmiş ve veri gizleme yöntemleri birbirleriyle karşılaştırılmıştır.

Son olarak, seçilen FPGA geliştirme kartı üzerinde üç farklı veri gizleme örneği gerçekleştirilmiş ve uygulama sonuçları LCD ekran üzerinde gösterilmiştir.

Anahtar Kelimeler: FPGA, veri gizleme, steganografi, spartan 3e geliştirme kartı

2013, ix + 62 sayfa

ABSTRACT

MSc Thesis

DATA HIDING APPLICATIONS WITH FPGA

Zeynep PEKER

Uludag University

Graduate School of Natural and Applied Sciences Department of Electronics Engineering

Supervisor: Asst. Prof. Dr. Ersen YILMAZ

The purpose of this master thesis is to perform a data hiding application with a FPGA platform. Xilinx Spartan 3E Development Kit was selected as a FPGA and 2*16 line LCD on the FPGA development card was used for display purposes.

In this thesis, we initially mentioned the historical development, the features and the advantages of FPGAs. Additionally, the properties and the advantages of VHDL ,which is a hardware description language and used in this study, the basics of FPGA programming and Xilinx ISE Design Studio 12.1 programme were explained in detail. Then, the history and the sub-branches of data hiding applications and the importance of steganography were summarized and the data hiding methods were compared with each other.

Finally, three different data hiding examples were implemented on the selected FPGA development card and the results were displayed on the LCD.

Key words: FPGA, data hiding, steganography, spartan 3e development kit

2013, ix + 62 pages

ÖNSÖZ VE/VEYA TEŞEKKÜR

Yüksek lisans eğitimim ve tez dönemim boyunca her zaman yanımda ve destekçim olan danışman hocam sayın Yrd.Doç.Dr. Ersen YILMAZ' a katkılarından dolayı teşekkürlerimi sunarım.

Hayatımın tüm aşamalarında kararlarımı destekleyen, yardımlarını esirgemeyen ve beni bu günlere getiren sevgili anneme ve her zaman yanımda olan kardeşlerime teşekkür ederim.

Zeynep PEKER

14/02/2013

İÇİNDEKİLER

Sayfa

ÖZET……..i

ABSTRACT……..ii

ÖNSÖZ ve TEŞEKKÜR……..……iii

SİMGE ve KISALTMALAR DİZİNİ……..vi

ŞEKİLLER DİZİNİ……..vii-viii

ÇİZELGELER DİZİNİ……ix

1. GİRİŞ…….1

2. KAYNAK ARAŞTIRMASI……..2

2.1. FPGA Üreticileri…….3

2.2. FPGA Yapısı……...4

2.2.1. Mantık hücresi (Logic Cell)……..5

2.3.VHDL Donanım Tanımlama Dili……6

2.3.1. Donanım tanımlama dili (Hardware description language - HDL)……...6

2.3.2. VHDL……..7

2.3.3. Tasarım akışı……8

2.3.4.VHDL temel yapıları……..10

2.4. FPGA Programlama……12

3.MATERYAL VE YÖNTEM……...15

3.1. Gizli Kanallar (Covered Channels)……..17

3.2. Gerçek Kimliği Saklama (Anonymity)……17

3.3. Telif Hakkı İşaretlemesi (Copyright Marking)……18

3.4. Steganografi (Steganography)……..19

3.4.1. Steganografi ile kriptografinin karşılaştırılması……....19

3.4.2. Steganografi ile damgalamanın karşılaştırılması……...…20

3.4.3. Steganografi nedir?.....21

3.4.4. Steganografinin tarihçesi……...22

3.4.5. Steganografinin alt alanları……23

3.4.6. Steganografinin kullanım alanları ……24

3.4.7. Metin (text) steganografi …….....25

3.4.8. Görüntü (Image) steganografi……..…….28

3.4.9. Ses (Audio) steganografi……...…31

3.4.10. Kullanılan diğer ortamlar…….32

3.5.Proje oluşturma ve sentezleme …….34

3.5.1.Programın FPGA'ya gömülmesi……45

4. FPGA ÜZERİNDE STEGANOGRAFİ UYGULAMASI……49

4.1. Spartan 3E Geliştirme Kartı Üzerindeki LCD Ekranı Sürme……...49

4.2.FPGA Üzerinden VHDL Donanım Tanımlama Dili LCD Ekran Sürme……..54

4.3. LCD Üzerinde Steganografi Algoritması…….55

6. SONUÇ……58

KAYNAKLAR……...…59-61

ÖZGEÇMİŞ…….62

SİMGELER ve KISALTMALAR DİZİNİ

HDL	Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VHDL	Vhsic Hardware Description Language
FPGA	Field Programmable Gate Array
LUT	Look Up Table
PROM	Programmable Read Only Memory
EPROM	Erasable Programmable Read Only Memory
ASIC	Application-Specific Integrated Circuit
CPLD	Complex Programmable Logic Device

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1. Gömülü sistem teknolojisinin yıllara göre gelişimi…… 2

Şekil 2.2. FPGA genel yapısı…… 4

Şekil 2.3. Mantık hücresi…….. 5

Şekil 2.4. VHDL tasarım adımları blok şeması…….8

Şekil 3.1. Spartan 3E geliştirme kartı (starter kit)…….16

Şekil 3.1. ISE yeni proje oluşturma…….17

Şekil 3.2. Proje isimlendirme……..18

Şekil 3.3. Proje özelliklerinin seçilmesi……..19

Şekil 3.4. Proje özetinin görüntülenmesi…….19

Şekil 3.5. Projeye VHDL modüle eklenmesi……..20

Şekil 3.6. VHDL modüle isimlendirme……...21

Şekil 3.7. VHDL modüle giriş çıkış port bilgileri……...21

Şekil 3.8. VHDL modüle açılış sayfası……...22

Şekil 3.9. Örnek VHDL kodu…….23

Şekil 3.10. Ucf dosyası ekleme……...23

Şekil 3.11. ucf dosyası isimlendirme……...24

Şekil 3.12. ucf dosyasının proje sayfasında görünümü……...25

Şekil 3.13. Örnek ucf dosyası kodu……25

Şekil 3.14. Projenin derlenmesi……..27

Şekil 3.15. IMPACT ara yüzüne erişim……..29

Şekil 3.16. bit uzantılı dosyanın FPGA’ ya aktarımı……..30

Şekil 3.17. Programın FPGA’ya yüklenmesi……..31

Şekil 3.18. Programın doğru şekilde yüklenmesi……32

Şekil 4.1. Bilgi gizleme yöntemlerinin sınıflandırılması…….32

Şekil 4.2. Resmin tamamına filigran gömülmüş örnek……...34

Şekil 4.3. Sayısal steganografi yöntemlerinin sınıflandırılması……..40

Şekil 4.5. Kelime kaydırma kodlaması örnek…….…….42

Şekil 4.6. Gelecek kodlaması örnek……...…….42

Şekil 4.7.Steganografik sistem……44

Şekil 5.1. Spartan 3E FPGA ile LCD arası bağlantı……49

Şekil 5.2. LCD şematik bağlantıları……50

Şekil 5.3. LCD veri yazma protokolü……..52

Şekil 5.4. LCD sürmek için örnek VHDL kod başlangıcı……...54

Şekil 5.5. LCD için ucf dosyası……..55

Şekil 5.6. Spartan 3E geliştirme kartı……..56

Şekil 5.7. Gizli veriyi içeren ekran görüntüsü…….57

Şekil 5.8. Gizli verinin çözülerek ekrana gönderilmesi……...57

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 5.1. LCD bitlerinin durumuna göre modlar……52

Çizelge 5.2. LCD Karakter Seti…….. 53

1. GİRİŞ

Bilgi ve teknoloji çağı olarak adlandırılan son yıllarda sürekli olarak yeni teknolojiler gündeme gelmektedir. Hayatımızın her alanında meydana gelen bu hızlı değişimlerden biri de gömülü sistem teknolojisinde yaşanmaktadır.

Programlanabilir lojik ürünler sayısal devrelerin tasarlanmasında uzun bir dönemden beri önemli kolaylıklar sağlamaktadır. 1970 yılında PROM teknolojisi ile başlayıp, 1980'lerde Boole fonksiyonlarının gerçekleştirilmesi için kullanılan PAL tüm devrelerinin gelişmesi ile devam eden teknoloji bu tür devrelerdeki lojik kapı sayısının giderek artması sonucunu oluşturmuştur.

Günümüzde programlanabilir lojik devrelerin en önemli uygulamalarından biri FPGA' lardır. Bu teknoloji benzer ürünlerine göre daha yüksek hız, kapasite ve esneklik özelliğine sahiptir. Ayrıca, hücreler arasındaki iletişim mimarisi bakımından da diğer teknolojilere üstünlük sağlamaktadır.

1984 yılında ilk defa Xilinx firması tarafından piyasaya sürülmüş olan FPGA' lar, daha sonra Actel, Altera, Atmel, Chip Express, Clear Logic, Cypress, DynaChip, Fast Analog Soltions Ltd, Gatefield, HenmerGres, Lattice, Lucent Technnologies, Military Aerospace Applications of Programmable Devices and Technologies Conference(MAPLD), Motorola, Orbit, Quicklogic, QuickTurn Vantis gibi firmalar tarafından üretilmeye başlanmışlardır. Ancak bu firmalar arasında en büyük pazar payına Xilinx ve Altera sahiptir.

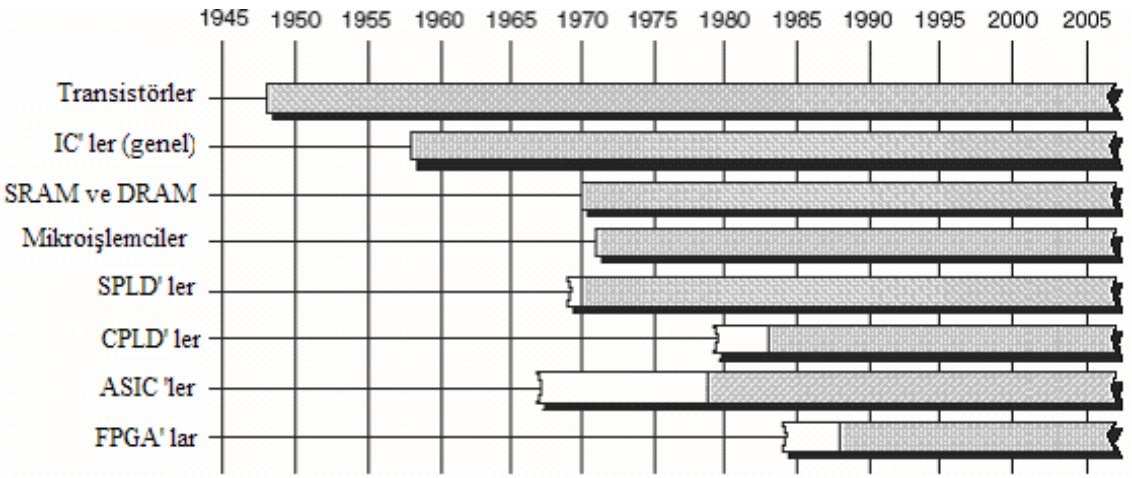
Gömülü sistem teknolojisinin yanı sıra bilginin güvenli bir şekilde iletilmesi gün geçtikçe önem kazanmakta ve gizli veri aktarımı için yeni yöntemler geliştirilmektedir. Steganografi veri gizlemenin en önemli alt dallarındandır ve resim, metin, ses üzerine uygulamaları yapılmaktadır. Eski çağlardan bu yana kullanılan steganografi bilimi sayesinde gizlenmek istenen bir bilgi üçüncü kişilerin fark edemeyeceği şekilde masum görünümlü bir medya ortamında taşınır.

Tez çalışmasında FPGA teknolojisi ile metin steganografi uygulamaları gerçekleştirmek hedeflenmiştir. Spartan 3E FPGA geliştirme kartı üzerinde belirlenen bir metin gizleme algoritması ile steganografi uygulamaları tamamlanmıştır.

2. KAYNAK ARAŞTIRMASI

FPGA (Alan Programlanabilir Lojik Kapı Dizileri) yapılandırılabilir mantık blokları ile bu bloklar arasında değiştirilebilir ara bağlantılardan oluşan yarı iletken sayısal tümleşik devrelerdir. Mantıksal bloklar basit lojik kapıların (AND, OR, XOR yerine vb.) işlemlerini yerine getirmek için programlanabildikleri gibi daha karmaşık fonksiyonların (şifreleme, görüntü işleme algoritmaları, matematiksel fonksiyonlar yerine vb.) işlemlerini de gerçekleştirebilirler. FPGA, üretim aşamasının ardından donanım yapısı yani iç konfigürasyonu kullanıcının gerçekleştirmek istediği fonksiyon doğrultusunda değiştirilebilen entegre devre olarak da tanımlanabilir.

Kullanıma başlandığı 1980' lerin ortalarında FPGA yongalarından veri işleme, matematiksel fonksiyonlar ve cihazlar arası bağlantı uygulamalarında yararlanılmıştır. 1990' ların ilk yıllarında artan kapasiteleri sayesinde geniş veri işlemleri gerektiren haberleşme ve ağ ortamlarında kullanımı artmıştır. 90' ların sonlarına doğru tüketiciye yönelik otomotiv ve endüstriyel uygulamalardaki kullanımları büyüme göstermiştir. 2000' lerin ilk yıllarında milyonlarca kapı içeren yüksek performanslı FPGA' lar piyasaya sürülürken, bazı tipleri gömülü mikroişlemci çekirdekleri, yüksek hızlı giriş/çıkış (I/O) arayüzleri, gömülü RAM ve DSP öbekleri sunmaktadır.



Şekil 2.1. Gömülü sistem teknolojisinin yıllara göre gelişimi (fpganedir.com 2012)

Şekil 2.1' deki beyaz renkli çubuklar tekniğin bulunduğu fakat bu zaman zarfında uygulamasına geçilmediğini göstermektedir.

2.1. FPGA Üreticileri

Dünyanın en büyük FPGA üretici firmaları Xilinx ve Altera' dır.

FPGA' in mucidi olan Xilinx firması şu anda dünyanın en büyük FPGA ve CPLD(programlanabilir platformlar) üreticisidir. 1984 yılında ABD' de kurulan firmanın dünya çapındaki pazar payı %50' nin üzerinde olup üç bin civarında çalışanı bulunmaktadır.

- Xilinx FPGA Ailesi :
- Virtex Ailesi : Yüksek performans gerektiren uygulamalar için üretilen FPGA ürün grubudur.
- Spartan Ailesi : Yüksek adetlerde üretime girecek uygulamalar için önerilen, daha ekonomik FPGA ürün grubudur.

- Xilinx ISE Design Studio :

Kapsamlı bir program olan Xilinx ISE Design Studio, Xilinx' in kendi FPGA ve CPLD' leri için sunduğu tasarım yazılımıdır. VHDL ve Verilog dillerini desteklemekte, şematik çizimlere imkan vermektedir. Ücretsiz versiyonu olan ISE WebPack Xilinx' in sitesinden indirilebilmektedir.

- Xilinx Platform Studio :

Xilinx' in kendi soft işlemcilerini istenen konfigürasyonda oluşturup projelere eklemek için sunduğu programdır. Platform Studio' da oluşturulan bir işlemci ISE' de oluşturulan bir projeye eklenebilir.

- MicroBlaze :

Xilinx' in kendi FPGA' leri için sunduğu orta performans / orta maliyet sınıfı 32 bit'lik soft (FPGA' e gömülebilen) işlemcisidir.

- PicoBlaze :

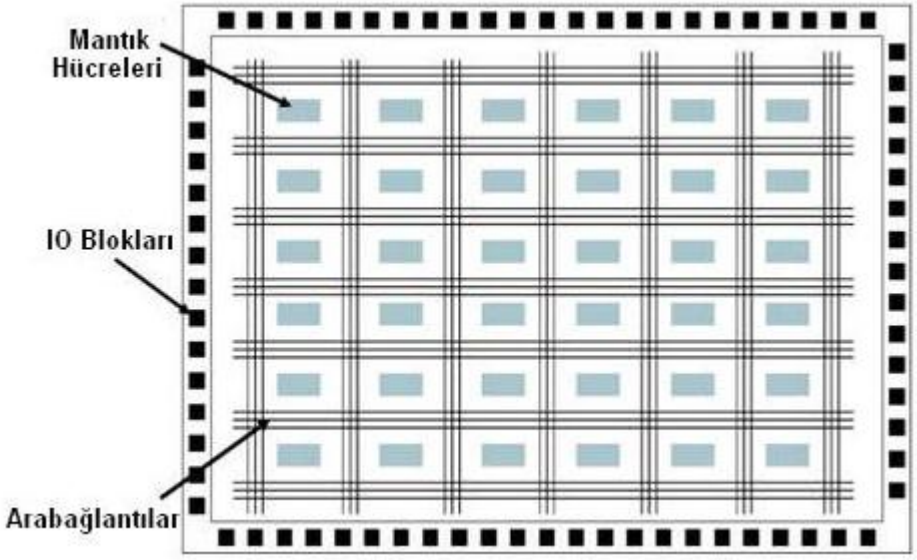
Xilinx' in kendi FPGA' leri için sunduğu düşük performans / düşük maliyet sınıfı 8 bit' lik soft (FPGA' e gömülebilen) işlemcisidir.

Altera dünyanın önde gelen firmalarından bir tanesidir. Firma FPGA tasarımı için QUARTUS yazılımını geliştirmiştir. Quartus programında özel tasarlanmış SOPC Builder kullanarak, FPGA içerisine soft mikroişlemci gömülebilmektedir. NIOS

yazılımı ile de C veya C++ kullanılarak mikroişlemci programlanabilmektedir. Bu iki programında ücretsiz versiyonları Altera' nın web sayfasından edinilebilmektedir.

2.2. FPGA Yapısı

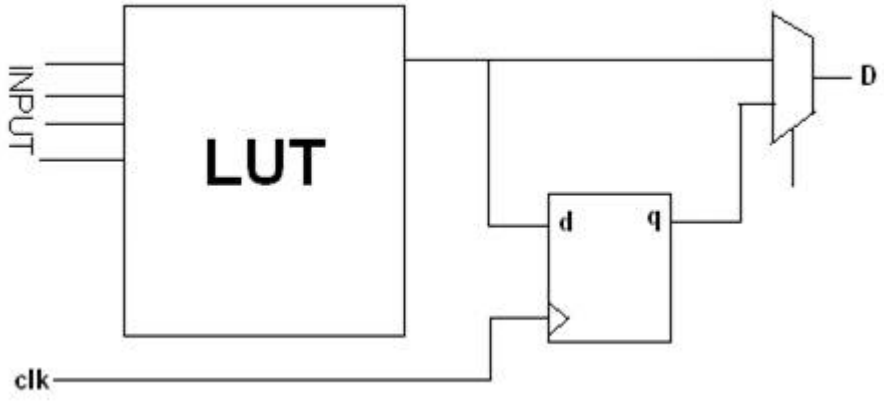
FPGA temel olarak mantık hücreleri (logic cell), giriş/çıkış blokları (I / O Block) ve ara bağlantılardan oluşur.



Şekil 2.2. FPGA genel yapısı (fpganedir.com 2012)

2.2.1. Mantık hücresi (Logic Cell)

FPGA' in ana yapısını mantık hücreleri oluşturur. Bir mantık hücresini 1 adet Lookup Table (LUT), 1 adet D Flip – Flop ve 1 adet 2 giriş – 1 çıkış mux oluşturur.



Şekil 2.3. Mantık hücresi (fpganedir.com 2012)

LUT' lar bir mantık işlemi yerine getiren küçük belleklerdir, n girişli bir LUT 2 e^n kapasiteli bir belleğe işaret eder. Binlerce mantık hücresinin birleşimi sonucu kompleks ve büyük programlar oluşur.

Mantık hücrelerinin ara bağlantıları matris şeklindeki veri yolları ve programlanabilir anahtarlarla sağlanır. FPGA tasarımı, her bir mantık hücresinin uygulayacağı fonksiyonu ve programlanabilir anahtarların durumunu (açık/kapalı) belirleyerek bu mantık hücreleri arasındaki bağlantıları tanımlar.

FPGA Pinleri

FPGA pinleri genel olarak 2 kategoriye ayrılır:

- Ayrılmış Pinler (Dedicated pins)
- Kullanıcı Pinleri (User pins)

Ayrılmış Pinler: Bir FPGA' de tüm pinlerin %20 ila %30' u ayrılmış pindir. Bu pinler, FPGA' de gerçekleştirdikleri özel fonksiyonlara göre 3' e ayrılır.

- Güç Pinleri: FPGA için gerekli olan güç ve toprak (ground) sağlayan pinlerdir.
- Konfigürasyon Pinleri: Oluşturulan programın FPGA' e yüklenmesi için kullanılan pinlerdir.

- Clock Pinleri: Clock sinyalleri için ayrılmış özel pinlerdir.

Kullanıcı Pinleri: Bu pinler kullanıcı tarafından konfigüre edilebilen standart I/O pinleridir. Giriş (input), çıkış (output) ve giriş/çıkış (input/output) olarak 3 farklı şekilde kullanılabilir. Her bir I/O pini FPGA' de bir IO hücresine bağlıdır, bir grup IO pini 2,5 V ile çalışırken bir grup pin 3,3 V ile çalışabilmektedir.

2.3.VHDL Donanım Tanımlama Dili

2.3.1. Donanım tanımlama dili (Hardware description language - HDL)

Donanım tanımlamak için HDL çok sayıda kapıdan oluşan sayısal sistemleri tanımlamada en etkili ve en hızlı yoldur. HDL ile donanım tanımlama dilinin sebepleri şöyledir:

- Kompleks sistem tasarımlarında geleneksel tasarım araçları yetersiz kalmaktadır.
- Eleman sayısındaki yoğunluk gün geçtikçe artmaktadır.
- Tasarım süresinin kısa olması gerekmektedir.
- Tasarlanmış bir modül sentezleme işlemiyle kapı eşdeğerine dönüştürülebilmektedir.

Genel olarak bir tasarım, HDL kodlarından ve bir Core Generator makrosundan oluşur. Yazılan HDL kodlarının, her bir programın kendi derleyicisi ile (Xilinx ISE, Quartus…) derlenmesi aşamasından sonra tasarımın doğrulanma aşamasında ise genellikle Modelsim simülatörü kullanılır. Tasarım, Xilinx programının kendi derleyicisi olan XST, Leonardo Spectrum ve ISim kullanılarak da sentezlenebilir.

Bu dillerden bazıları VHDL, Verilog ve Abel' dir. Verilog VHDL' e göre öğrenilmesi ve program geliştirilmesi daha kolay olan bir donanım tanımlama dilidir. Akademik çalışmalarda ve kompleks tasarımlarda tercih edilen donanım tanımlama dili ise VHDL' dir.

2.3.2. VHDL(Very High Speed Integrated Circuit Hardware Description Language)

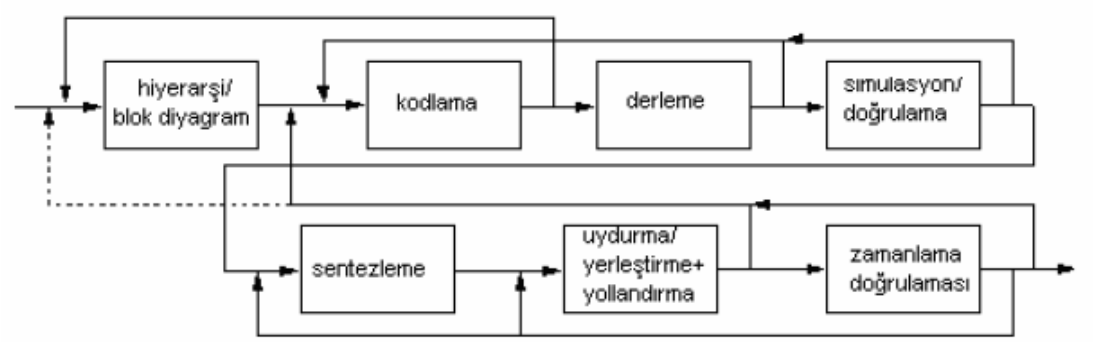
VHDL, IEEE (IEEE – standart 1076) ve Birleşik Devletler Silahlı Kuvvetleri (United States Military Department) tarafından standart bir tanımlama dili olarak kabul edilmiştir. Bu standart birkaç kez revizyondan geçtikten sonra 1987 versiyonu en sık kullanılan versiyon olmuştur. Bu dil vasıtası ile donanım katmanları tanımlanır, aynı zamanda dokümantasyon, doğrulama ve büyük sayısal sistemlerin sentezlenmesi sağlanır. VHDL' nin özellikleri aşağıdaki gibidir:

- Tasarımlar hiyerarşili şekilde bileşenlerine ayrılabilir.
- Her bir tasarım elemanı iyi bir arayüze (diğer elemanlarla bağlantı için) ve hatasız davranışsal tanımlamaya sahip olmalıdır.

- Davranışsal tanımlama yapılırken algoritma veya gerçek donanım yapıları kullanılarak elemanların işlemi belirtilir.
- Uyumluluk, zamanlama ve saatle denetim modellenebilir. VHDL senkron ve asenkron ardışıl devre yapılarını gerçekleyebilir.
- Bir tasarımın lojik işlemleri ve zaman davranışının simülasyonu yapılabilir.

2.3.3. Tasarım akışı

VHDL tabanlı tasarım işlemi birkaç adımda yapılabilir, bu adımlar HDL tabanlı tüm tasarımlara uygulanabilir. Şekil 2.4' de tasarım adımlarının taslağı görülmektedir.



Şekil 2.4. VHDL tasarım adımları blok şeması (ce.istanbul.edu.tr/erdem/courses 2012)

İlk olarak yapılacak tasarımın temel yaklaşımları ve blok diyagram seviyesinde tasarımı yapılır. Büyük sayısal tasarımlar genel olarak hiyerarşik bir yapıya sahiptir.

Bir sonraki adım, modüllerin VHDL kodlarının yazılmasıdır. Bu kodlama işleminde modüllerin ara yüzleri ve yapıları detaylı bir şekilde yazılır. VHDL, metin temelli bir dil olduğundan bu işlemi yapmak için metin editörü kullanılabilir.

Kod yazım aşamasından sonra, doğruluğunu kontrol etmek için derleme işlemi yapılmalıdır. VHDL derleyicisi, yazıların kodların söz dizim kurallarına uygunluğunu ve diğer modüllerle uyumunu analiz eder. Derleyici aynı zamanda simülasyon işleminde kullanılacak dosyaları da oluşturur.

Derleme işlemi hatasız olarak yapıldıktan sonra bir sonraki adım olan simülasyon aşamasına geçilir. VHDL simülatörü, tasarımdaki girişleri uygulamayı ve çıkışları gözlemlemeyi herhangi bir fiziksel devreye sahip olmadan gerçekleştirir. Simüle edilen devrenin istenen çıkışı elde ettiği doğrulanır. Burada fonksiyonel ve zamanlama olmak

üzere iki farklı doğrulama vardır. Fonksiyonel simülasyonda, zamanlamadan bağımsız olarak devrenin sayısal işlemi üzerinde durulur. Kapı gecikmeleri ve diğer zamanlama parametreleri sıfır olarak düşünülür. Zamanlama doğrulamasında, devrenin tahmini yaklaşık gecikmesi ve diğer zamanlama gereksinimleri incelenir. Burada elde edilen zamanlama sonuçları tahmini değerlerdir. Gerçek sonuçlar sentezleme ve yerleştirme sonuçlarına bağlıdır.

Simülasyon aşamasında doğrulama yapıldıktan sonra sentezleme aşamasına geçilir. Bu adımda VHDL tanımları, donanımsal olarak hedef teknolojiye uygulanır. Sentezleme işlemi kullanılan araçlara ve hedef teknolojiye göre farklı şekillerde yapılır. Devre gerçekleştirilirken çalışılacak üretici (Xilinx, Altera, Atel…vs) seçilir. Sentezleme araçları, kullanılacak kapıların listesini ve bunlar arasındaki arabağlantıları belirten bir bağlantı listesi (netlist) üretir.

Sentezleme sırasında elde edilen bağlantı listesi yerleştirme adımında yerleştirme araçları tarafından kullanılır. Bu araçlar sentezlenmiş bileşenleri hedef devreye yerleştirir. Bu adımda tasarımcı modüllerin yerleşimi, dış giriş ve çıkış pinlerinin atanması gibi ek kısıtlamalar tanımlayabilir.

En son aşamada yerleştirilen devrenin zamanlama doğrulamasının yapılmasıdır. Sadece bu adımda, kablo uzunluğu, elektriksel yükleme ve diğer faktörlerden kaynaklanan gerçek devre gecikmesi hesaplanabilir. Eğer hesaplanan değerler isteği karşılıyorsa tasarım tamamlanır.

Tasarım yapılırken adımlar arasında ileri ve geri gitme olabilir. Örneğin, kodlama sırasında herhangi bir problemle karşılaşılırsa geriye dönülüp hatalar tekrar incelenebilir. Simülasyon sonucu istenildiği gibi değilse kodlama işlemi gözden geçirilir.

VHDL kullanımının en büyük avantajları şu şekilde sıralanabilir:

- Tasarım tanımlamalarının bağımsız olması
- Birçok üreticinin kullanabilmesi
- Teknolojinin gerisinde kalan elemanların kullanımının engellenmesi
- Tasarımın güncellenmesi

- Tasarımın süresinin kısalması
- Ürünün pazara hızlı çıkma imkanının olması
- Tasarım maliyetinin azalması
- Tasarım kalitesinin artması
- Fonksiyonların yüksek seviyede kontrolünün yapılabilmesi
- Tasarım elemanlarının tekrar kullanımını sağlar.

2.3.4.VHDL temel yapıları

Sayısal bir sistemin VHDL tanımlaması yapılırken dört temel yapı kullanılır. Bu yapılar entity bölümü, mimari bölüm, konfigürasyon ve paketlerdir.

Sayısal bir sistem, modüllerin bir hiyerarşi ile birleştirilmesiyle oluşur. Her modül VHDL' de bir entity ile ifade edilir. Her entity ile giriş çıkışları ve fonksiyonu tamamen belirlenmiş olan bir donanım yapısı gösterilir. Her tanımlamanın entity ve mimari olmak üzere iki bölümü vardır. Entity bildirimi tasarımın dış bağlantılarının, mimari bölümü ise içeride yapılacak işlemlerin gösterilmesinde kullanılır. Pakette pek çok tanımlamada ortak kullanılacak genel bilgiler verilir. Konfigürasyon ise yapısal tanımlamada kullanılacak alt sistemlerin giriş çıkış bilgileri tanımlanır.

- Entity Bildirimi:

Entitiy bildiriminde bir modülün bağlantıları ve bağlantı yolları türü tanımlanır. Fakat bunların arasındaki ilişki konusunda bilgi verilmez. Yazılım kuralı şöyledir:

entity modülün ismi **is**

[jenerik_bildirimi]

[giriş/çıkış tanımları]

{entity_bildirim_elemanları}

[**begin**

devrenin çalışma şartlarının kontrol edildiği bölüm]

end [modülün ismi];

Generic_bildirimi modülün yapısını veya davranışını kontrol etmek üzere kullanılan sabitlerin tanımlandığı alandır. Yazılış şekli aşağıda verilmiştir:

generic (

sabit ismi : tipi [: = ilk değeri]

{ ; sabit_ismi : tipi [: = ilk değeri]});

Giriş/çıkış tanımları modülün bağlantı yollarının tanımlandığı alandır ve yazılış şekli aşağıdaki gibidir:

port (

port_ismi : [mod] tipi [: = ilk değeri]

{; port_ismi : tipi [: = ilk değeri]}) ;

Giriş/çıkışlar modülün çevre modüllerle haberleşmesini sağlayan yollardır. Her giriş/çıkış bir mod(in, out, inout, buffer) ve bir veri tipi ile gösterilir. Giriş/çıkışların dört modu şu şekildedir :

- **in** : sadece okunabilir, sadece giriş için kullanılır.
- **out** : sadece bir değer verilebilir, sadece çıkış olarak kullanılır.
- **buffer** : okunabilir ve değer yazılabilir. Devrenin içinden sadece bir sürücüsü olabilir.
- **inout** : okunabilir ve değer yazılabilir. Devrenin içinden birden fazla sürücüsü olabilir.

➢ Mimari bildirim:

Mimaride modülün girişleri ile çıkışları arasındaki ilişkiler tanımlanır. Tanımlama davranışsal, veri akışı veya yapısal olmak üzere üç farklı şekilde yapılabilir. Bir mimaride aşağıdaki yazım şekli kullanılır:

architecture mimari_ismi **of** modülün_ismi **is**

{mimari_bildirim_bölümü**}**

begin

{ eşzamanlı_satırlar }

end [mimari_ismi] ;

- Davranışsal Mimari :

Davranışsal mimaride sistemin yaptığı işlem process' ler kullanılarak program benzeri bir şekilde yazılır, fakat tasarımın nasıl gerçekleneceği detaylandırılmaz. Bir mimaride birden fazla process varsa hepsi eş zamanlı çalıştırılır, fakat process' ler içindeki satırlar yukarıdan aşağıya doğru sırayla çalıştırılır.

- Veri Akışı Mimari:

Veri akışı mimaride, kontrol işaretleri ve verilerin, toplayıcı, karşılaştırıcı, kod çözücü ve basit lojik kapılar gibi kombinasyonel devre elemanları üzerindeki eş zamanlı hareketleri tanımlanır.

- Yapısal Mimari :

Yapısal mimaride eş zamanlı çalışan alt modüllerin listesi ve birbirlerine bağlantıları tanımlanır. Tasarımın karmaşıklığı arttıkça bütün tasarımı bir defada tanımlamak oldukça zorlaşır. Bu sebeple sistem daha az karmaşık alt sistemlere ayrılır. Her t sistem tek başına gerçekleştirilebilen veya alt sistemlere ayrılabilen birimlerdir. VHDL' de yapısal mimari kullanılan yüksek seviyedeki tasarım, sadece daha önce tanımlanmış alt sistemlerin listesi ve birbirleriyle olan bağlantılarından oluşur (Öcal 2006).

2.4. FPGA Programlama

FPGA programlama dört aşamadan oluşmaktadır:

- Sayısal Tasarım:

Sayısal tasarım aşamasında, tasarım bir şematik tasarım editörü ile veya bir Hardware Description Language (HDL) ile oluşturulur. Şematik giriş programı devrenin grafik sembollerini kullanır. HDL giriş programında ise devre tanımlayıcı bir dil kullanılır (Örneğin Verilog, Abel, VHDL…). Şematik giriş imkanı sağlayan programlar, deneyimli tasarımcıların aygıt üzerindeki bölümlerin saptanmasında ve yerleştirilmesinde daha fazla kontrol imkanı sağlar. HDL programlarında ise tasarım giriş aşaması oldukça hızlıdır. Fakat kısa zamanda tasarım sağlama avantajının yanında, performans kaybını ve yoğunluk dezavantajını da getirmektedir.

Özellikle FPGA tasarım konusunda son yıllarda HDL programları oldukça gelişmiştir. Her iki yöntemde de aygıtın mimarisini ve özelliklerini bilmek daha etkili tasarım yapılmasını sağlar. Cihazın teknolojisini göz önünde bulundurmadan tasarım yapmak mümkün olsa bile genellikle performans kaybı ve verimsiz yer kullanım sonuçları oluşur.

- Tasarım Yorumlama ve Eşleme:

Sayısal tasarım aşamasında, tasarım programının üretmiş olduğu netlist formatındaki dosya FPGA yongasını ayarlayan uygun bit formatına dönüştürülür. Bu aşama üç alt bölümden oluşmaktadır.

Birinci aşama bölümlemedir, yapılmış olan tasarım FPGA yonga üzerindeki kaynaklar ile eşleştirilir. Bu aşamanın iyi bir şekilde yapılması FPGA için önemlidir, ne kadar iyi bir bölümleme sağlanırsa yönlendirme daha verimli kullanılır ve daha çok işlev sığdırılabilir.

İkinci aşamada, birinci aşamada yapılan eşleştirmeler doğrultusunda lojik blokların atanmasını ve yerleştirilmesini yapar. Bu aşamada hedef lojik blokların yerlerinin, yönlendirmenin minimum olmasını sağlayacak şekilde tespit edilmesidir. Böylelikle sistem performansında kayda değer bir artış sağlanır. Bu aşama, FPGA ve büyük CPLD tasarımları için büyük hesaplama gücü gerektirir. Bu aşamada yorumlama yazılımı, blokları yerleştirirken yol uzunluklarını ve tıkanıklıklarını takip eder. Bazı yazılımlar kullanıcı tarafından belirlenen zaman limitlerini karşılayabilmek için bekleme sürelerini de takip eder.

Üçüncü aşamada, lojik bloklar arasındaki yönlendirmeler ve bağlantılar tanımlanır. Bu aşamada çıktı, mantık hücre dizisi (Logic Cell Array, LCA) formatındaki FPGA yongaya mahsus bir dosyadır. Bu LCA dosyası daha sonra FPGA yongayı ayarlamak için uygun bit sırasına çevrilir.

Büyük veya karmaşık uygulamalarda, yazılım başarılı bir şekilde yerleştirme ve yönlendirme yapamayabilir. Bazı yazılım paketleri tam yönlendirme yapabilmek için farklı seçenekleri deneyen yazılımlar içerir. Genel olarak, aygıt üzerindeki kaynakların %85' inden azının kullanılması tavsiye edilir. Bu teknik, yazılıma rahat yönlendirme yapılabilmesi için fazladan kaynaklar sunar. Bazı üreticiler, fiziksel yerleştirmeye

müdahele etmek için araçlar sağlar. Yerleştirme büyük FPGA' ler için oldukça önemlidir. Çünkü bazı yazılımlar tasarım yapısını kavramayabilir. İyi bir yerleştirme yazılımı tasarım yapısını yerleştirip yönlendirmeyi yapabilmesini sağlar.

- Tasarım Kontrol ve Benzeşim:

Tüm tasarım aşamalarında tasarım doğrulama yapılmaktadır. Bu aşamada yapılmış olan tasarım test sinyalleri ile, devre lojiğini ve zamanlamasını test eder. Programlanabilir lojiğe uygulanan bazı temel testler vardır. Fonksiyonel simülasyon tasarım giriş aşamasında kontrol edilmektedir, yerleştirme ve yönlendirme yapılmadan önce lojik doğruluğu kontrol edilir. Tam zaman testi, yerleştirme ve yönlendirmeden sonra yapılabilir. Yerleştirme ve yönlendirmeden sonra, zamanlama simülasyonunu uygulayarak, yollardaki ve lojik bloklardaki bekleme sürelerini hesaplar. Simülasyon hep tavsiye edilmekle beraber, kapı dizilerinde olduğu gibi yoğun simülasyon aşamalarına ihtiyaç yoktur. Devre içi doğrulama testi ise başka bir kontrol yöntemidir, tasarımı normal çalışma koşulları içinde kontrol eder. Böylece yonga çalışırken zamanlama değerleri kontrol edilir. Yapılabilecek iyileştirmeler yonga çalışırken yongaya yansıtılır ve yeni değerler doğrultusunda iyileştirme en iyiyi elde edene kadar devam eder.

- FPGA Ayarlama ve Çalıştırma:

Bu aşama bit dosyasının FPGA içine yüklendiği aşamadır. FPGA yonganın ayarlanması bit dosyasının tipine bağlı olarak değişir. FPGA yongalar seri PROM yongalar vasıtası ile programlanabilir. FPGA program uçucu özellikte olduğundan programı kendi üzerinde saklamaz. FPGA' in elektriksel bağlantısı kesildiğinde program kaybolur. FPGA' in iki modu vardır:

- Konfigurasyon Modu: FPGA güç verildiğinde konfigurasyon moduna geçer. Bu modda FPGA bekleme durumundadır ve bütün çıkışları pasiftir. FPGA programlaması bu aşamada yapılır.
- Kullanıcı Modu :Yükleme bittikten sonra FPGA kullanıcı moduna girer ve pinleri aktif hale gelir. Böylece FPGA kendine verilen görevi yerine getirmeye başlar.

3. MATERYAL VE YÖNTEM

Tez çalışmasında materyal olarak Spartan 3E FPGA geliştirme kartı ve steganografi bilim dalı kullanılmıştır.

Spartan 3E Starter Kit içerisinde;

- Geliştirme kartı,
- Üniversal güç kaynağı,
- Değerlendirme yazılımı (Xilinx ISE Web Pack),
- El kitabı,
- Starter Kit kaynak CD' si,
- USB Kablosu

bulunmaktadır.

Geliştirme kartı üzerinde bulunan donanımlar;

Xilinx araçları:

- Spartan - 3E FPGA(XC3S500E-4FG320C)
- CoolRunner – II CPLD (XC2C64A – 5VQ44C)
- Platform Flash (XCF04S-VO20C)

Saat İşaret Kaynağı:

- 50 MHz kristal saat osilatörü

Bellek:

- 128 Mbit paralel flash
- 16 Mbit SPI Flash
- 64 MByte DDR SDRAM

Konnektör ve arayüzler:

- Ethernet10/100
- JTAG USB arayüzü
- İki adet 9 pin RS - 232 seri port
- PS/2 fare/klavye portu
- 4 adet kayan anahtar
- 8 adet LED çıkışı

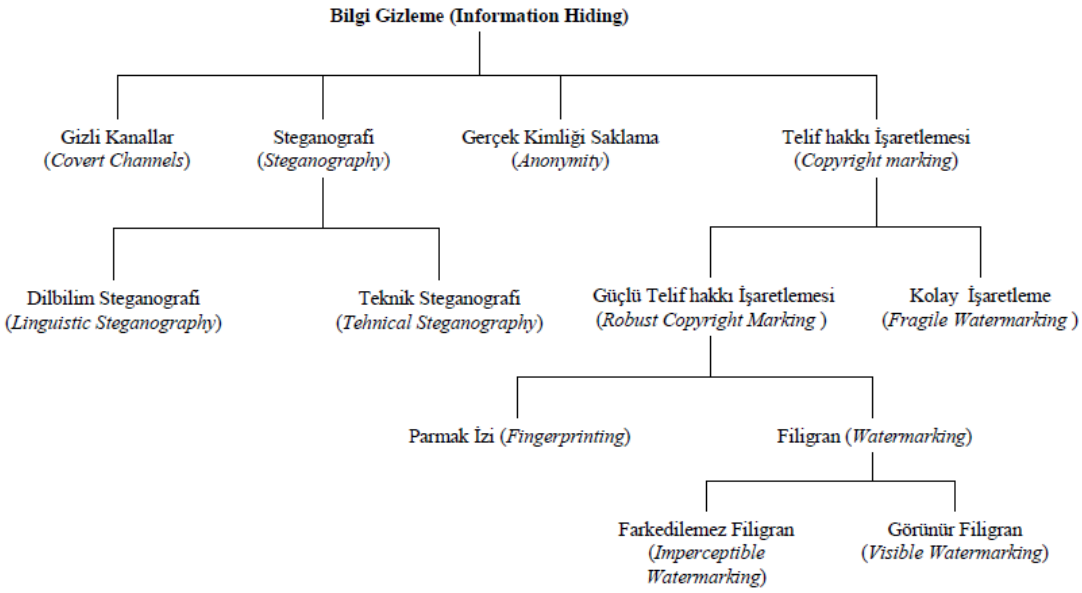
- 4 adet anlık temaslı buton
- 100-pin genişleme bağlantı portu
- 3 adet 6-pin genişleme konnektörü

Ekran:

- 2 satır – 16 karakter LCD Ekran

Şekil 3.1. Spartan 3E geliştirme kartı (starter kit) (User Guide 2011)

İletişimin hızlı bir biçimde gerçekleştirilebilmesi için çeşitli yöntemler geliştirilmiştir. Bilgi gizlemenin sınıflandırılması aşağıda gösterilmiştir. (Pfitzmann, 1996). Bu sınıflandırma, bilgi gizleme konusunda yapılan ilk çalıştayda üzerinde çalışılarak kabul edilmiştir (Anderson, 1996).



Şekil 3.2. Bilgi gizleme yöntemlerinin sınıflandırılması (Şahin, 2007)

3.1. Gizli Kanallar (Covered Channels)

Bilgi gizlemenin ilk alt disiplini olan gizli kanallar Lampson (Lampson, 1973) tarafından tanımlanmıştır. Gizli kanallar iki kişi arasında gizli bilgilerin el değiştirmesi için iletişimi sağlayan kanaldır. Gizli kanal kurulması iki kişinin karşılıklı anlaşmasını gerektirmektedir. Gizli kanalların amaçları, iletişimimizdeki veriyi saklamaya çalışmak ve iletişiminin amacını saklamaktır. Böylece; gerçek veri transferi, dikkatsiz gözlere zararsız ve kanuna uygunmuş gibi gözükecek ve veriyi karıştırmak için ayrı bir şifreleme yapılmasına gerek kalmayacaktır.

3.2. Gerçek Kimliği Saklama (Anonymity)

Diğer bir alt alan olan gerçek kimliği saklama, veri gönderimi sırasında gerçek kimliği saklayarak, bilginin bilinmeyen ya da anlaşılamayan biri üzerinden gidiyor olduğunu izlenimi verilerek gönderilmesidir. Bu şekilde bilgi zarar görmeden gönderilebilmektedir. Fakat ağlar üzerinde bilinmeyen kullanıcı olayı ağ yöneticilerinin daha fazla dikkatini çekmekte ve bilgi güvenliği tehlikeye girmektedir. Bu yüzden

sadece çok gerektiği durumlarda kullanılması uygundur (Chaum, 1981) (Goldschlag vd.,1996).

3.3. Telif Hakkı İşaretlemesi (Copyright Marking)

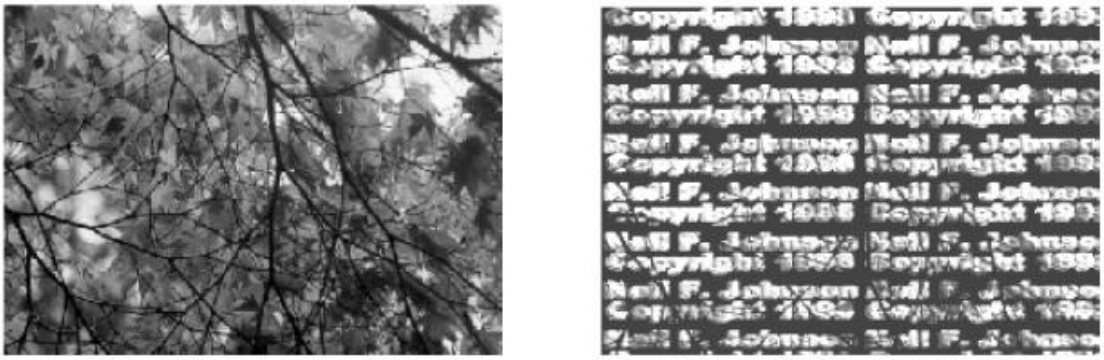
Telif hakkı işaretlemesinde ise orijinal dosyanın korunması amacıyla dosyanın içine bazı bilgiler gizlenmektedir. Bunlar; dosyaların üretildiği tarih, telif hakkı sahibi, üreticiye nasıl ulaşılabileceği gibi bilgileri içermektedir. Bu yöntemler steganografi ile beraber kullanılmaktadır. Telif hakkı işaretlemesi, sayısal görüntülerde sayısal filigran olarak kullanılmaktadır (Swanson vd., 1998). Filigran, bir çeşit gizli damga baskısıdır. Örneğin kâğıt banknotlar üzerindekiler gibi. Bunlar ancak ışığa tutularak bakıldıklarında görülebilmektedirler. Modern steganografi uygulamalarında kullanılan filigranlar ise görüntü ve ses dosyalarında kopyalamayı önlemek amacıyla damgalar bırakmaktadırlar [Hartung ve Kutter, 1999]. Bu damgalar özel programlar tarafından okunabilmekte ve dosyaların üretildiği tarih, telif hakkı sahibi, üreticiye nasıl ulaşılabileceği gibi bilgileri içermektedir.

Filigran ile korunmuş görüntüler parlaklık ve zıtlık (kontrast) ayarlarının değiştirilmesi, özel filtrelerin kullanılması, kâğıda baskı ve tarama gibi birçok yönteme karşı koyabilmektedir. Fakat gelişen teknolojiyle birlikte ortaya çıkartılan bazı yeni programlar kullanılarak bu filigran aşılabilmektedir. Sayısal filigranlar ikiye ayrılır. Bunlar;

1. Görünür filigran (visible watermark)
2. Görünmez filigran (invisible watermark)

Görünen filigranlar insan gözünün rahatlıkla görebileceği izlerdir. Örneğin paralar da bulunan ışığa tutunca görünen resimler (TL. deki Atatürk resmi gibi), bir başka örnek ise televizyon kanallarında o görüntünün hangi kanal yada ajans tarafından çekildiğini gösteren ekranın köşesinde bulunan bir logodur. Bir görünür filigrana saldırı, ancak o kısmın kesilerek çıkarılmasıyla yapılabilir.

Görünmeyen filigrana bir örnek ise; pasaportlarda bulunan kişiye ait seri numarasının fotoğrafın içerisine de gömülmesidir. Herhangi biri elde ettiği bir pasaporta kendi resmini yapıştırdığı zaman özel tarayıcılarla fotoğrafı tarandığında seri numarasının tutmadığı ya da olmadığı gözükecektir. Görünmeyen filigranların görünen filigranlara göre bazı avantajları vardır. Filigran yerleri belli değildir ya da filigran olup olmadığı fark edilmeyebilir. Filigranı tüm resim içine dağıtmak genel bir uygulamadır (Şekil 1.2).



Şekil 3.3. Resmin tamamına filigran gömülmüş örnek (Johnson vd. 2000)

Bu durum resmi kesme saldırılarına (cropping attacks) karşı biraz olsun koruma sağlar. Fakat dosya içerisine gömülecek olan bilgi ne kadar az ise saldırılara karşı o kadar güçlü ve güvenli olur. Bu dosya içerisindeki tekrarlılığın (redundancy) azalması için gereklidir (Johnson vd., 2000).

3.4. Steganografi (Steganography)

Steganografi bilgi gizleme yöntemlerinin en önemli alt dalıdır (Petitcolas vd., 1999). Bu yaklaşım, bir nesnenin içerisine bir verinin gizlenmesi olarak tanımlanabilir. Dilbilim ve teknik steganografi olarak ikiye ayrılmaktadır. Bu yaklaşımla ses, sayısal resim, video görüntüleri üzerine veri saklanabilir. Görüntü dosyaları içerisine saklanacak veriler metin dosyası olabileceği gibi, herhangi bir görüntü içerisine gizlenmiş başka bir görüntü dosyası da olabilir.

3.4.1. Steganografi ile kriptografinin karşılaştırılması

Steganografinin kriptografiden (şifreleme) en önemli farkı steganografide saklı mesajın varlığının gizlenmesidir. Yani saklı verinin örtü verisi içine gömüldüğü bilgisi sadece mesajın alıcısı tarafından bilinir ve örtü verisine sahip olan bir başkası saklı verinin varlığını fark edemez. Kriptografide ise gönderilen verinin gizli olduğu herkes tarafından bilinir. İçeriği gizli anahtar olmadan anlaşılamaz ve gizli verinin anlaşılabilmesi için çok büyük çabanın ve zamanın harcanması gerekir. Kriptografide güçlü algoritmaların kullanılması nedeniyle deneme (brute-force) saldırılarına karşı dirençli olup gizli verinin elde edilmesi çok güç olmaktadır. Analiz için güçlü bilgisayarlara ihtiyaç duyulmaktadır.

Steganografide ise mesajın herhangi bir nesnede saklandığı anlaşıldığında gizli veriyi elde etmek nispeten kolaydır. Steganografi sistemlerinde mesaj iletilirken içindeki bilgi kriptolanabilir böylece üçüncü bir kişi tarafından mesajın varlığı tespit edilse bile gizli anahtar olmadığından mesaj hala gizliliğini koruyacaktır. Ancak haberleşmenin gizliliğinin fark edilmesi nedeniyle steganografi esas amacına ulaşamamış olacaktır.
Ayrıca kriptografide kullanılan algoritmaların çoğu herkes tarafından bilinmektedir. Ancak steganografi yöntemleri hâlihazırda geliştirilmeye açıktır ve çalışmalar devam etmektedir.

3.4.2. Steganografi ile damgalamanın karşılaştırılması

Steganografide amaç gizli verinin fark edilmemesi olmasına karşın damgalamada bu söz konusu değildir. Damgalamada örtü nesnesinin korunması amaçlanmaktadır. Diğer bir deyişle gizlenecek veri, örtü nesnesi için bir güvenlik sağlamaktadır.
Damgalamanın steganografiden bir diğer farkı; damgalamada gizli verinin ne kadar gömüldüğünden çok gizli verinin saldırılara karşı dayanıklı olması amaçlanmaktadır. Önemli olan örtü nesnesinin tabi olacağı saldırılara rağmen gizli verinin geri alınması sonrasında gizli veride herhangi bir değişikliğe uğramadan elde edilebilmesidir. Damgalamada kullanılacak alan genelde kısıtlı olduğundan damgalama kapasitesi de bu oranda küçük olacaktır. Steganografide gizli mesajın gömülmesiyle örtü nesnesinde fark edilmeyecek derecede de olsa bozulmaya izin verilirken, damgalama uygulamalarında gizli mesaj örtü verisinin bir parçası olarak saklanır ve genellikle örtü verisinin bozulmasına izin verilmez.

Ayrıca damganın analizciler tarafından kaldırılamaması, insan gözüyle algılanamaması, istatistiksel olarak anlaşılamaması, JPEG gibi kayıplı sıkıştırmalara karşı dayanıklı olması, kes-yapıştır işlemleri gibi saldırılara dayanıklı olması damgalama yöntemlerinin sahip olması gereken özelliklerindendir.

3.4.3. Steganografi nedir?

Steganografi eski bir bilgi gizleme sanatıdır (Petitcolas vd., 1999). Steganografi kelimesi kökleri "*στεγανος*" ve "*γραΦειν*"'den gelen Yunan alfabesinden türetilmiştir. Tam olarak anlamı "*kaplanmış yazı*" (*covered writing*) demektir (Murray ve Burchfiled, 1933). Steganografi' nin amacı gizli mesaj ya da bilginin varlığını saklamaktır. Taşınmak istenen mesaj bir başka masum görünüşlü ortamda saklanarak, üçüncü şahısların iletilen mesajın varlığından haberdar olması engellenir. Metin, ses, sayısal resim, video dosyaları üzerine veri saklanabilir. Bu veriler metin dosyası olabileceği gibi, herhangi bir görüntü içerisine başka bir görüntüyü gizlemekte olasıdır. Yine aynı şekilde bir ses dosyasının içine bir metin dosyası da saklanabilmektedir (Memon ve Wong, 1998) (Wang ve Wang, 2004). Steganografi gizli bir iletişim sağlamaktadır. Amacı iki kişi arasındaki iletişimin bir üçüncü şahıs tarafından fark edilememesidir. Bilimsel ortamda Steganografi çalışmaları 1983 yılında Simmons tarafından "Prisoner Problem"'in (Simmons, 1984) tanımlanması ile başlamaktadır. Bu problemde Alice ve Bob hapishanededir ve hapishaneden kaçmak için planlar yapmaktadırlar. Fakat bu planların gardiyan Willie'ye fark ettirilmeden yapılması gerekmektedir. Eğer Willie bunu fark ederse kaçma planları suya düşecektir. Bu nedenle de çeşitli gizli haberleşme yöntemleri geliştirilmesi gerekmektedir.

Bu yaklaşımda içine bilgi gizlenen ortama örtü verisi (cover-data) veya örtü nesnesi (cover-object), oluşan ortama da stego-metin (stego-text) veya stego-nesnesi (stego object) denmektedir (Kharrazi vd., 2004).

Gizlenecek Mesaj + Örtü-nesnesi = Stego-nesnesi

Steganografi şifrelemeye yakın olmasına rağmen şifrelemeden faklıdır. Şifreleme mesajın içeriğinin korunması ile ilgilenirken steganografi mesajın varlığının gizlenmesi ile ilgilenmektedir. Dolayısıyla steganografi bir şifreleme yöntemi değil şifrelemeyi tamamlayıcı bir öğedir (Anderson ve Petitcolas, 1998).

3.4.4. Steganografinin tarihçesi

Steganografi, Antik yunan ve Heredot zamanına kadar uzanan oldukça eski bir veri gizleme yöntemidir.

Yunan tarihçi Heredot, eserinde (Heredotus, M.Ö. 430), İran'da bulunan casusun, Pers istilasını Yunanistan'a nasıl ilettiğini kaydetmektedir. Yazıya göre, casus kölesinin saçını kazıtmış; istila uyarısını da kafa derisine kazıtmıştır. Daha sonra yapılacak olan, kölenin saçının yazıyı kapatacak kadar uzamasını beklemek ve bu köleyi Yunanistan'a göndermektir. Kölenin bilmesi gereken tek bilgi *"kafamı kazıyın"* olacaktır. Yine aynı çağda avcı kılığındaki bir ulağın, avladığı hayvanın karnına parşömen saklayarak Yunanistan'a girmesi anlatılmaktadır (Newman, 1940). Antik çağda steganografinin kullanımı yalnızca Yunanistan ile sınırlı değildir. Çinliler de kendi kaynaklarında meyve sepetini nasıl gizli iletişim için kullandıklarını anlatmaktadırlar. Meyve sepetindeki her meyvenin birbirine göre pozisyonu farklı bir anlam ifade edecektir. Antik dönemdeki bu basit uygulamalar steganografinin gizli iletişimdeki kullanımının insanlık kadar eski olduğunu bizlere göstermektedir (Tacticus, 1990). Steganografi hakkında yazılan ilk kitap Johannes Trithemus (1462–1516) tarafından yazılmış olan *Steganographiæ* isimli kitaptır. 1600'lü yıllarda yaşamış olan Gaspar Schott (1608–1666) tarafından yazılmış olan *Schola Steganographica* (Schott, 1665) isimli kitapta ise müzik notalarının bilgi gizlemek için nasıl kullanıldığı anlatılmıştır. Bu yöntem birçok bilgi gizleme yöntemine de temel oluşturmuştur. Daha sonraki yıllarda steganografi, görünmez mürekkep, metin belgelerindeki harf frekanslarını kullanma, I. ve II. Dünya Savaşlarında kullanılan mors kodları gibi uygulamalarla karşımıza çıkmaktadır (Katzenbeisser ve Petitcolas, 2000). Ancak, çarpıcı kullanımı ikinci dünya savaşında kendini göstermektedir. İkinci dünya savaşı esnasında, Alman casusların gizli bilgileri kimyevi bir madde ile beyaz bir mendile yazdıkları ortaya çıkartılmıştır. Casus, gizli mesaj içeren bu mendili daha önce belirlenen noktalarda çöpe atmakta; alıcı ise yine kimyevi maddeler kullanarak bu yazıyı okumaktadır (Kahn, 1967).

Yine ikinci dünya savaşı döneminde Almanlar "mikrofilm" teknolojisi kullanarak "mikro noktalar" (microdot) kullanmıştırlar. Bu yöntemde A4 büyüklüğündeki herhangi bir belge veya çizim bir dizi işlem sonrasında daktilo yazısında kullanılan bir nokta kadar küçültülmektedirler. Bu yöntem kullanılarak masum içerikli bir sayfa düz metindeki i ve j harflerinin noktalarına oldukça büyük miktarda veri saklamak mümkün

olmuştur (Zim, 1948). Aşağıda, ikinci dünya savaşında kullanılan bir steganografi örneği verilmiştir (Johnson ve Jajodia, 1998).

"*Apparently neutrals protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretex for embargo on by-products, ejecting suets and vegetable oils*."

Yukarıda verilen paragrafta her kelimenin ikinci harfleri yan yana getirildiğinde "*Pershing sails from NY June 1*." Mesajı ortaya çıkmaktadır. Günümüzde sayısal (dijital) nesneler üzerinde steganografi uygulamaları yapılmaktadır ve gelişen teknoloji nedeniyle, verilerimizi korumak amacıyla son yıllarda sıklıkla kullanılmaya başlanmıştır. Gizli veri, yine masum içeriğe sahip olan bir dizi dosyanın içinde saklanabilmektedir. Bunlardan en ilgi çekicileri, vermiş oldukları olanaklardan dolayı, resim, ses ve video dosyalarıdır. Benzer bir şekilde düz metin dosyaları, sabit disklerdeki kullanılmayan alanlar, IP (Internet Protocol) paketlerinin ileride kullanmak üzere ayrılmış bölümleri gizli verinin saklanması için kullanılabilmektedir. Html dosyaları, exe dosyaları vb. gibi dosyalar da içlerine veri saklamada kullanılabilmektedir.

3.4.5. Steganografinin alt alanları

Steganografi, Dilbilim Steganografi (Linguistic Steganography) ve Teknik Steganografi (Technical Steganography) olmak üzere kendi içerisinde ikiye ayrılmaktadır [Johnson ve Rude, 2001]. Dilbilim Steganografi, taşıyıcı verinin text olduğu steganografi koludur. Burada değişiklik yapmanın çeşitli yolları vardır. Bunlardan bazıları şöyledir:

− Grafik kullanılarak yapılabilir,

− text'in yapısı değiştirilerek yapılabilir

− ya da amacı sadece veriyi saklamak olan yeni bir text yaratılabilir.

Dilbilim Steganografi'de kullanılan yöntemler ise şunlardır:

− **Açık kodlar:** Gizli mesaj, açıkça okunabilir fakat zararsız bir mesaj haline gelir. Bu işlem; maskeleme, boş şifreler ve grid (ızgara) ile yapılmaktadır.

− **Şemagramlar:** Gizli mesaj, açık metinin ufak fakat gizli bir detayının içine gizlenmektedir. Bunun için grafiksel değişiklikler yapılmaktadır. Kullanılan yöntemler ise; farklı yazı tipleri kullanmak, eski daktilo yazılarını kullanmak, resimler içinde boşluklar kullanmak vb'dir.

Teknik Steganografi, birçok konuyu içine almaktadır. Bunları bazı başlıklar altında toplayabiliriz;

− **Görünmez mürekkep:** Geleneksel haline gelmiş olan görünmez mürekkeple yazma yöntemidir.

− **Gizli yerler:** Kimsenin göremeyeceği gizli yerlere saklama (bavul, kasa vb.)

− **Microdot'lar:** Bilgiyi noktalar halinde sayfaya gizleme

− **Bilgisayar tabanlı yöntemler:** Text, ses, görüntü, resim dosyalarını kullanarak veri gizleme yöntemleridir.

3.4.6. Steganografinin kullanım alanları

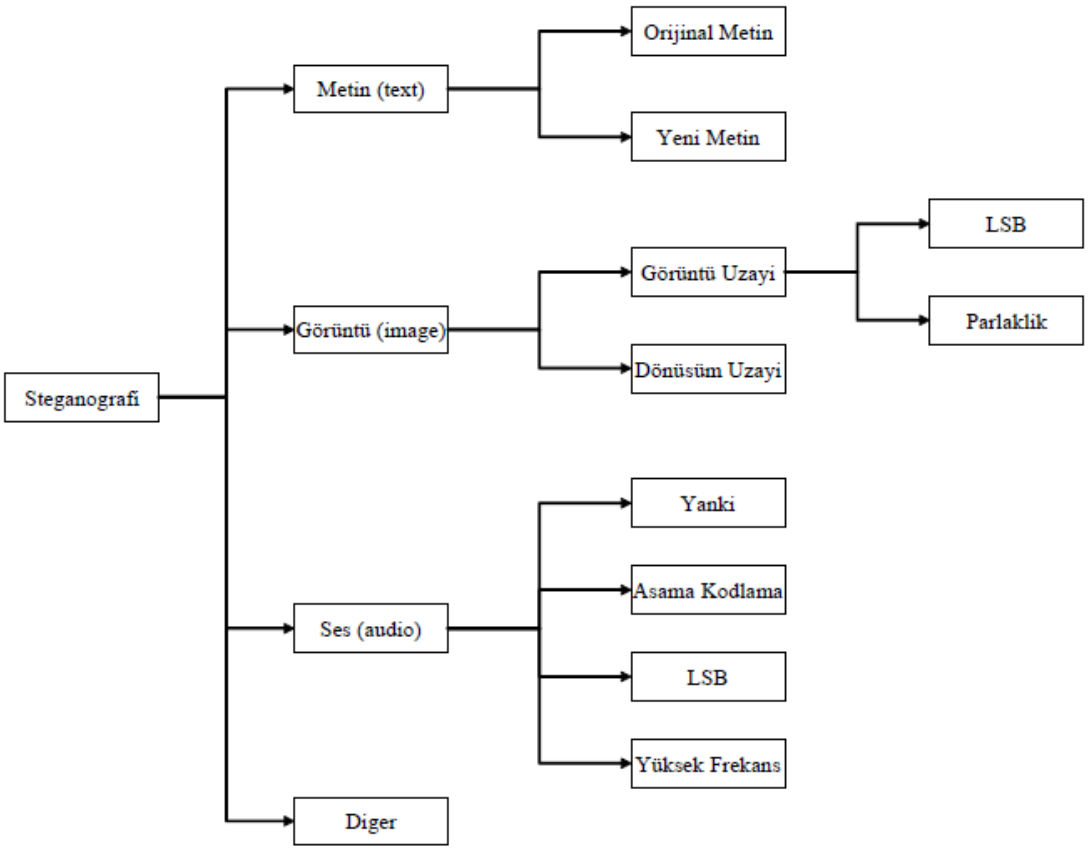
Sayısal steganografi kullanım alanları açısından genel olarak üçe ayrılmaktadır. Bunlar aşağıdaki gibidir:

− Metin (text) steganografi

− Görüntü (image) steganografi

− Ses (audio) steganografi

Yaygın olarak kullanılan sayısal steganografi yöntemlerinin sınıflandırılması Şekil 4.3'de verilmektedir.



Şekil 3.4. Sayısal steganografi yöntemlerinin sınıflandırılması (Şahin 2007)

3.4.7. Metin (text) steganografi

Metin steganografi bilgi gizlenecek ortamın metin (text) olduğu steganografi koludur. Metin steganografinin uygulanabilmesi için çeşitli yöntemler vardır. Bunlar şu şekilde sınıflandırılabilir (Popa, 1998);

− Açık Alan Yöntemleri (Open Space Methods)

o Satır Kaydırma Kodlaması

o Kelime Kaydırma Kodlaması

o Özellik Kodlaması

− Yazımsal Yöntemler (Syntactic Methods)

− Anlamsal Yöntemler (Semantic Methods)

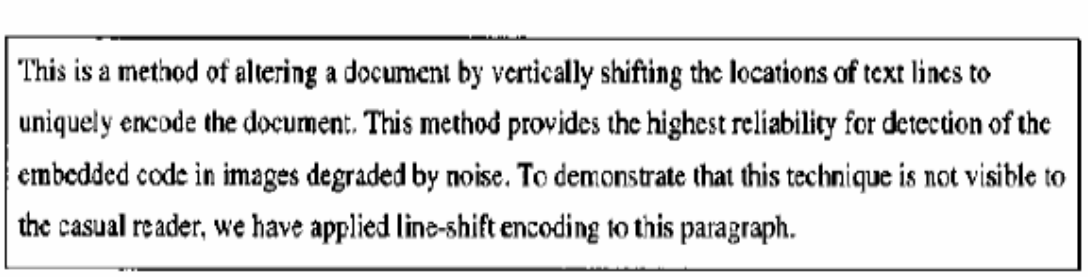
- Açık alan yöntemleri

Bu yöntemler, anormal gözükmeyen iki kelime arasında ekstra boşluklar ve satır sonu boşlukları ile çalışmaktadır. Bununla birlikte açık alan yöntemlerinin ASCII kodları ile

kullanılması daha uygundur. Açık alan yöntemlerinde kullanılan kodlama yöntemleri şu şekildedir.

- Satır kaydırma kodlaması

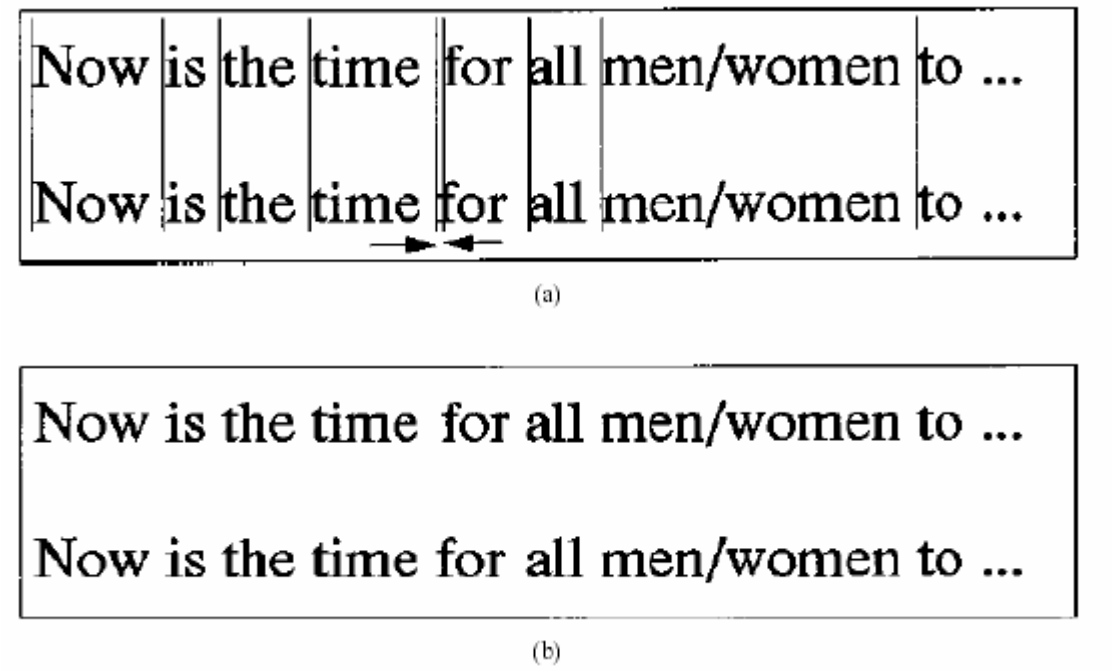
Bu yöntemde metin satırları düşey olarak kaydırılarak gömülecek mesajın kodlanması sağlanır. Gömülmüş kelime yine metin dosyası ya da Windows Bitmap (BMP) (Microsoft Corporation, 1990) dosya olarak açılabilir. Aşağıdaki metinde ikinci satır 1/300 inch yukarıya kaydırılmıştır. Fakat gözle anlaşılır bir fark yoktur. Bu yapılan "0" ya da "1" ile tanımlanarak kodlama işlemi gerçekleştirilir.


This is a method of altering a document by vertically shifting the locations of text lines to uniquely encode the document. This method provides the highest reliability for detection of the embedded code in images degraded by noise. To demonstrate that this technique is not visible to the casual reader, we have applied line-shift encoding to this paragraph.

Şekil 3.5. Satır kaydırma kodlaması örneği (Şahin 2007)

- Kelime kaydırma kodlaması

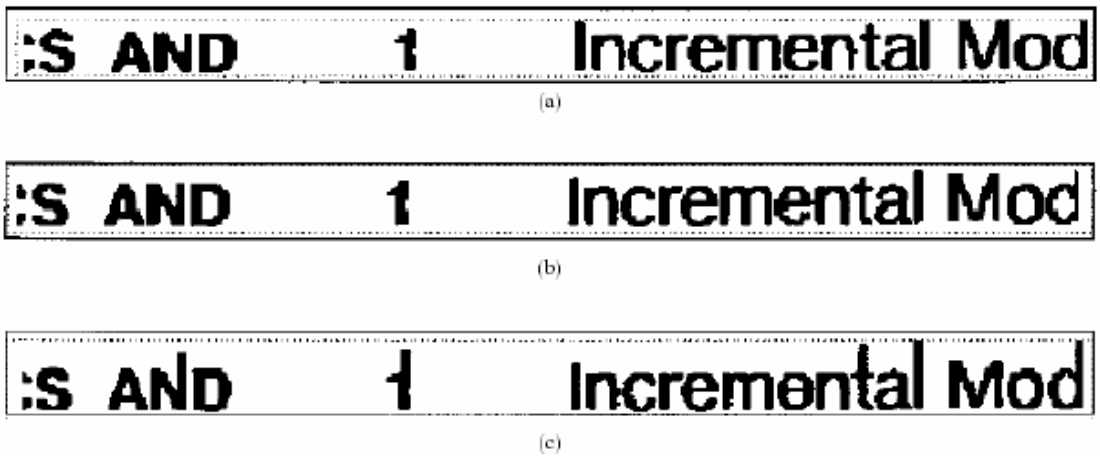
Bu yöntemde metnin satırları yatay olarak kaydırılarak dokümanın tek olarak kodlanması sağlanır. Gömülmüş kelime yine metin dosyası ya da BMP dosyası olarak açılabilir. Bu yöntem, dokümana uygulandığında yakın kelimeler arasında çok ta fark edilmeyen boşluklar ortaya çıkmaktadır. Bu oluşan boşluklardan dolayı dokümanın kodunun çözülmesi için eski belgeye de ihtiyaç vardır.



Şekil 3.6. (a) Üst satır'da "for" kelimesinden önce bir boşluk eklenmektedir, alt satırda for ile all arasında daha fazla boşluk vardır. **(b)** Dikey çizgiler olmadan metnin nasıl gözüktüğü (Brassil,1995)

- Özellik kodlaması

Bu kodlama tekniği hem metin belgelerine hem de bitmap dosyalara uygulanabilmektedir. Burada kelimelerin yerleri ve bazı harflerin boylarıyla oynanmakta ve ASCII kodlarında değişiklik yapılmaktadır.



Şekil 3.7. (a) Herhangi bir kodlama yapılmamış orijinal metin. **(b)** Sadece seçilen karakterler üzerinde yapılmış gelecek kodlaması. **(c)** Gelecek kodlamasının abartılmış gösterimi (Brassil,1995)

- Yazımsal yöntemler

Bu yöntem, dokumanı kodlamak için noktalama işaretlerini kullanır (Bender vd., 1996). Örneğin aşağıdaki iki cümle ilk bakışta aynıymış gibi gözükmektedir, fakat dikkatlice bakıldığında ilk cümlenin fazladan bir ‘,’ işareti içerdiği görülmektedir. Bu yapıların biri “1”, diğeri de “0” olarak belirlenmekte ve kodlama işlemi bu şekilde gerçekleştirilmektedir.

“bread, butter, and milk”

“bread, butter and milk”

- Anlamsal yöntemler

Bu yöntem W. Bender tarafından ortaya atılmıştır. Bu yöntemde eşanlamlı kelimelere birincil ve ikincil değerler atanmaktadır. Sonra bu değerler “1” ve “0” olarak binary’e dönüştürülmektedir. Örneğin *“big”* kelimesi birincil, *“large”* kelimesi de ikincil olarak işaretlenmiş olsun. Birincil “1”, ikincil de “0” olarak binary’e çevrilmektedir.

Bu uygulamaların dışında CSS ve HTML çalışmalarında da metin steganografi çalışılmaktadır (Kabetta vd., 2011).

3.4.8. Görüntü (Image) steganografi

Sayısal resimler dağıtımı en kolay ve internette hemen her sayfada karşılaşılabilecek dosyalardır. Kullanıldıkları formatlara göre farklılık göstermekle birlikte steganografi uygulamalarında en yaygın kullanılan ortamlar resim dosyalarıdır. Bu nedenle steganografi konusunda yapılan çalışmalar ve geliştirilen teknikler ağırlıklı olarak resim steganografi çerçevesinde yer almaktadır. Görüntü dosyalarının içerisine bir metin gizlenebileceği gibi bir resim dosyasının içine bir başka resmi de gizlemek mümkündür. Gizli bilgiyi bir resme gömme (ya da gizleme) işleminde iki dosya söz konusudur. Kapak resim ya da örtü verisi (cover image) olarak adlandırılan ilk dosya, gizli bilgiyi saklayacak resim dosyasıdır. İkinci dosya ise gizlenecek bilgi olan mesajdır. Bu mesaj da stego olarak isimlendirilmektedir. Mesaj, açık metin (plain text), şifreli metin (chipher text), başka resimler veya bit dizisi içinde saklanabilecek başka bir şey olabilir. Gömme işlemi sonucunda kapak resim ve gömülü mesajın oluşturduğu dosyaya “stego resim” adı verilir. Birçok farklı yöntem kullanılarak resimlerde bilgi gizlenebilmektedir.

Kullanılan yöntemler, gömme işlemi sırasında kullandıkları veri dikkate alınarak iki başlık altında toplanabilmektedir [Johnson ve Jajodia, 1998].

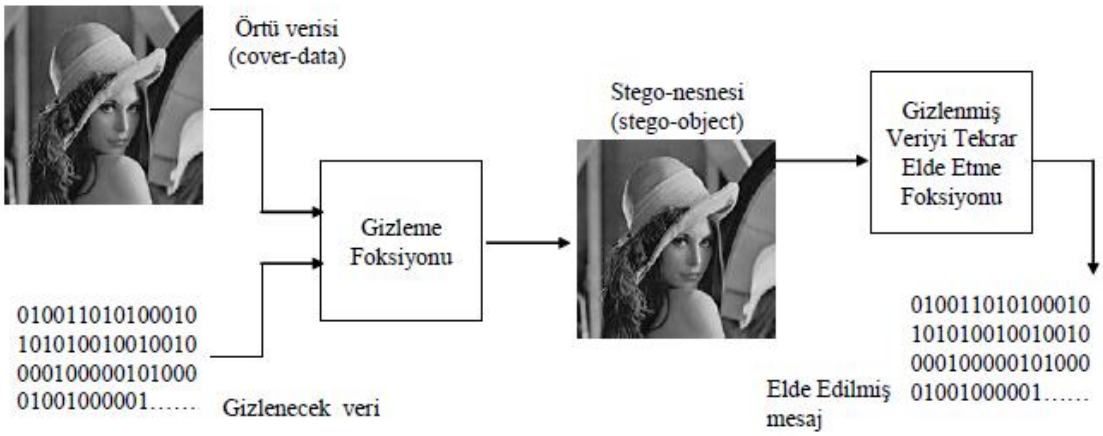
1. Uzaysal / Görüntü Alan Tekniği (Spatial / Image Domain Technique)
2. Frekans / Dönüşüm Alan Tekniği (Frequency / Transform Domain Technique)

Uzaysal Alan veya Görüntü Alan olarak adlandırılan teknik, gömme işleminde resim dosyasındaki veriyi doğrudan kullanılır. Gömme işlemin de bilgiyi gizlediği veri kümesi piksel değerlerini temsil eden kısımdır. Bu tekniğe örnek olarak yaygın olarak kullanılan En Önemsiz Bite Ekleme (Least Significant Bit Insertion - LSB) yöntemi gösterilebilir.

Frekans Alan veya Dönüşüm Alan olarak bilinen teknik ise kapak verideki değişimler üzerinde gömme işlemini uygular. Dönüşüm Alan tekniğine örnek olarak ise JPEG formatlı resim dosyalarına veri gömme işleminde kullanılan algoritmaları verebiliriz. Bu algoritmalar JPEG sıkıştırma sırasında kullanılan DCT katsayıları üzerinde veri gömme işlemini uygular.

Görüntü dosyaları için bir steganografik sistem Şekil 4.8.'de gösterilmektedir.

Gönderici bir gizleme fonksiyonu kullanarak bir steganogram yaratır. Gizleme fonksiyonu, verinin saklanacağı taşıyıcı ortam ve gizlenecek veri olmak üzere iki parametreye sahiptir (Westfeld ve Pfitzmann, 1999).



Şekil 3.8. Steganografik sistem (Şahin 2007)

Herhangi bir steganografik sisteminin temelde şeffaflık (transparency) ve sağlamlık (robustness) şartlarını sağlaması gerekmektedir. Şeffaflık saklanan verinin tespit

edilememesi ve fark edilememesini ifade ederken sağlamlık saklanan verinin çıkartma işleminde düzgün bir şekilde geri getirilmesini anlatmaktadır.

Görüntü steganografide, bilgilerin görüntü dosyaları içerisine saklanmasının çeşitli yöntemleri vardır. Şekil 4.8.'de Gizleme Fonksiyonu olarak adlandırılan ve bilgi gizlemede en çok kullanılan yöntemler aşağıda gösterilmiştir:

- En önemsiz bite ekleme
- Maskeleme ve filtreleme
- Algoritmalar ve dönüşümler (Sellars, 1999).

En önemsiz bite ekleme en yaygın kullanılan bilgi gizleme yöntemlerinden biridir. Taşıyıcı ortamın en az önemli bitlerini insan gözünün fark edemeyeceği şekilde gizli veriyi saklamak amacıyla değiştirmeyi temel alır.

Maskeleme ve filtreleme yöntemleri genellikle 24 bit resimler için kullanılmakta olup resmin en önemsiz alanlarının tespit edilerek buralarda saklama yapılmasını temel almaktadır. Bu yöntemler genelde filigran uygulamalarında karşımıza çıkmaktadır. Maskeleme teknikleri JPEG formatındaki resim dosyaları için daha uygundur. Dönüşümler ise yine daha çok JPEG dosyalar üzerinde kullanılmaktadır. En yaygın olarak kullanılan dönüşümler ise DCT ve DFT'dir.

3.4.9. Ses (Audio) steganografi

İnsan işitme sistemi (Human auditory system-HAS) frekans aralığı yüzünden, ses sinyalleri içerisine bilgi gizleme oldukça uğraş gerektiren bir konudur. HAS 1/1.000'den daha büyük frekans aralığını fark edebilir. Aynı zamanda HAS nereden geldiği belli olmayan gürültülere de oldukça duyarlıdır. Ses sinyalleri üzerinde uğraşırken ses dosyalarının hangi karakteristiklere sahip olduklarının bilinmesi gerekmektedir.

Ses dosyaları iki ana özelliğe sahiptirler:

− Basit nicelendirme metodu: Yüksek kaliteli sayısal seslerin 16-bit doğrusal nicelendirme ile ifadesinde en çok kullanılan yöntemdir. WAV (Windows Audio Visual) ve AIIF (Audio Interchange File Format). Bazı sinyal bozulmaları bu formatta ortaya çıkabilir.

− Geçici seçme oranı: Ses için en çok kullanılan oranlar 8 kHz, 9.6 kHz, 10 kHz, 12 kHz, 16 kHz, 22.05 kHz ve 44.1 kHz 'dir. Bu değer frekans aralığının kullanılabilecek en üst seviyesidir.

Bir diğer sayısal gösterim ise ISO MPEG-Audio formatıdır. Bu algılama ile ilgili bir formattır. Bu yöntemde sinyal istatistiği değiştirilir. Böylece ses korunur fakat sinyal değiştirilmiş olur [Sellars, 1999].

Ses dosyalarında veri gizleme yöntemleri ise şunlardır:

− Düşük bit kodlaması (Low-bit encoding)

− Aşama kodlaması (Phase coding)

− Taft yayılması (Spread spectrum)

− Yankı veri gizlemesi (Echo data hiding)

- Düşük bit kodlaması

Görüntü steganografide kullanılan LSB ekleme yöntemiyle aynı şekilde gerçekleştirilir. Ses dosyasındaki verinin her baytının son bitine gizlenecek bilginin bir biti yazılır. Sonuçta oluşan değişiklik ses dosyasında gürültüye neden olmaktadır. Ayrıca dayanıksız bir yapısı vardır. Tekrar örnekleme veya kanalda oluşabilecek gürültü ile mesaj zarar görebilir veya yok edilebilir (Kim vd., 2003).

- Aşama kodlaması

Aşama kodlaması yöntemi de resim dosyalarında uygulanan JPEG algoritması benzeri bir yapı taşımaktadır. Gömme işleminde ses dosyası küçük segmentlere bölünür ve her

segmente ait aşama (faz) gizlenecek veriye ait aşama referansı ile değiştirilir. Aşama kodlaması prosedürü aşağıdaki gibidir (Bender vd., 1996).

− Ses verisi N adet kısa segmente bölünür.

− Her segmente Discrete Fourier Transform (DFT) uygulanarak aşama ve büyüklük (magnitude) matrisleri yaratılır.

− Komşu segmentler arasındaki aşama farklılıkları hesaplanır.

− Her segment için yeni bir aşama değeri bilgi gizlenerek oluşturulur.

− Yeni aşama matrisleri ile büyüklük matrisleri birleştirilerek yeni segmentler elde edilir.

− Yeni segmentler birleştirilerek kodlanmış çıkış elde edilir.

- Taft yayılması

Gizleme işlemini ses sinyalinin kullandığı frekans taftı üzerinde yapmaktadır. Güçlü bir yapısı olamamakla birlikte seste gürültü meydana getirmektedir (Bender vd., 1996).

- Yankı veri gizlemesi

Bilginin gizlenmesi taşıyıcı ses sinyali üzerine bir yankı eklenmesi ile sağlanmaktadır. Bilgi yankının gecikme miktarı, zayıflama oranı veya büyüklüğü gibi değerler kullanılarak gizlenir. İki farklı gecikme değeri kullanılarak insan kulağının algılamayacağı düzeyde 0 veya 1'in kodlanması mümkündür. Her bitin kodlanması için sinyal segmentlere bölünür. Yankı veri gizlemesi yöntemi herhangi bir gürültüye neden olmamakta veya kayıplı bir kodlama kullanmamaktadır (Gruhl vd., 1996).

3.4.10. Kullanılan diğer ortamlar

Steganografi uygulamalarında yaygın olarak kullanılan text, görüntü ve ses dosyaları haricinde, sabit disklerdeki kullanılmayan alanlar, IP (Internet Protocol) paketlerinin ileride kullanmak üzere ayrılmış bölümleri gizli verinin saklanması için kullanılabilmektedir. Yine aynı şekilde Html dosyaları, exe dosyaları vb. gibi dosyalar da içlerine veri saklamada kullanılabilmektedir. Resim ve ses dosyalarına veri saklama yöntemleri insanın görme ve işitme sisteminin fark edemeyeceği ufak değişikliklerle bilgi gizleme mantığını temel almaktadır. Html ve exe gibi dosyala veri saklama yöntemleri ise bu dosyaların kendi formatlarındaki esneklikleri temel alarak çalışırlar.

Örneğin html dosyalarında etiketler (tag) kullanılmaktadır. Bu etiketlerin açma ve kapama şekilleri vardır. Bir metni şekillendiren iki etiket olduğunda bunları kaparken

hangisinin daha önce kapandığı hangisinin daha önce açıldığı sayfanın görünümünde fark oluşturmaz.

Örneğin:

<tr>deneme</tr> ile

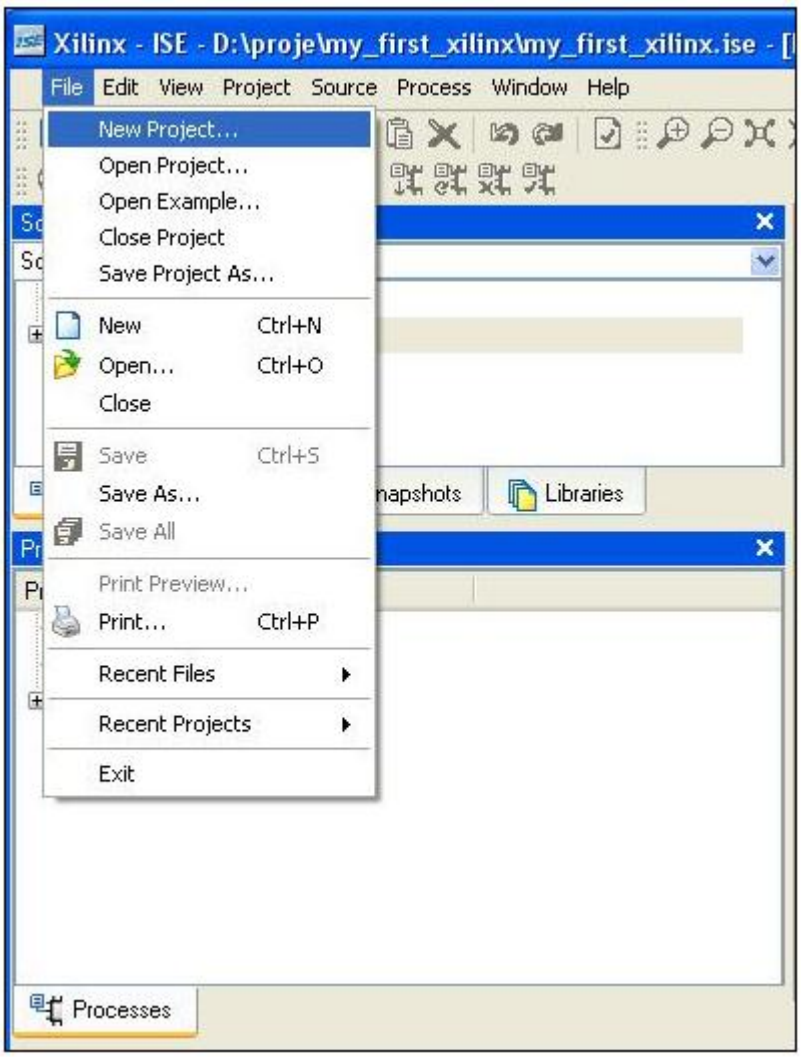
<tr>deneme</tr> satırları aynı görüntüyü sağlamaktadır.

Ayrıca html dosyasında arada bırakılan boş satırların sayısı görüntüyü değiştirmemektedir. Bu durumda, bu alanlar veri saklama amacıyla kullanılabilmektedir. Ancak bu yöntemlerin saklama kapasitesi düşüktür ve steganalitik yöntemlere karşı dayanıklılığı azdır.

Exe dosyalarında da benzer mantıkla hareket edilmektedir. Komut setlerinde aynı işlevi gören farklı komutlar olabilmekte ve bunlardan hangisinin kullanıldığı oluşan exe dosyasının çalışmasında değişikliğe neden olmamaktadır. Bu durum veri saklama amacıyla kodlanabilmektedir. Ancak yine burada da saklama kapasitesindeki düşüklük gündeme gelmektedir (Atıcı, 2005).

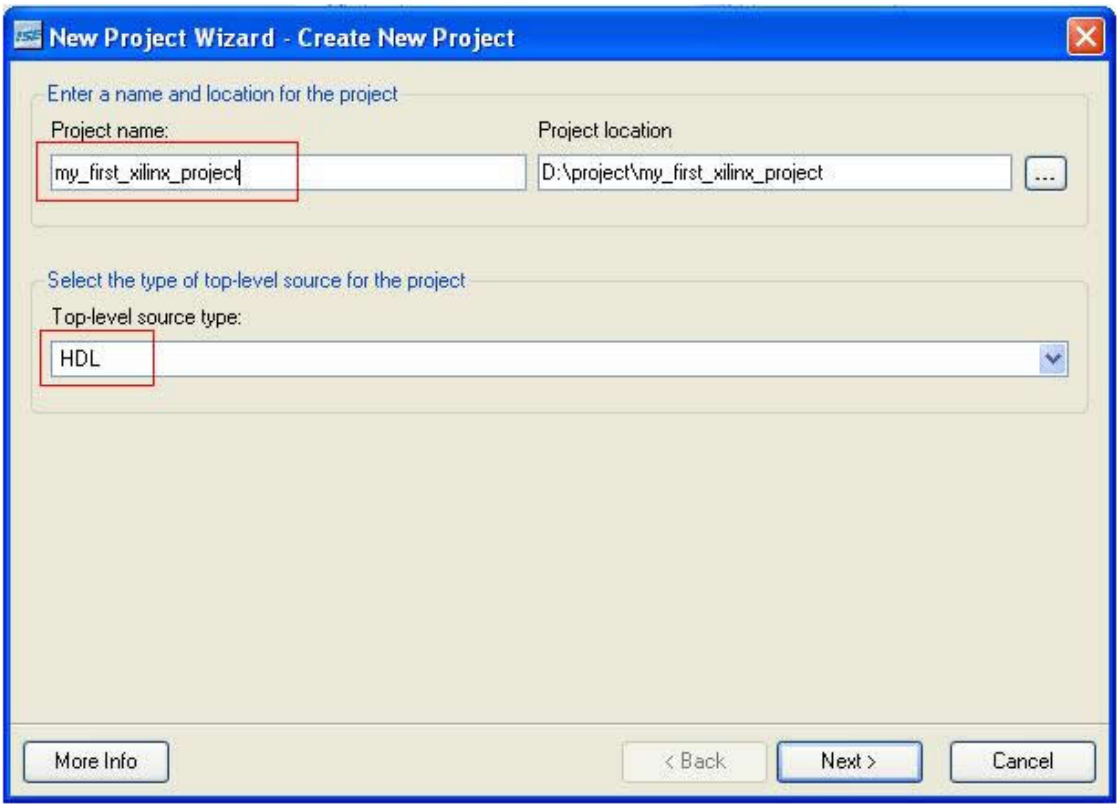
3.5. Proje oluşturma ve sentezleme

Xilinx ISE 12.1 Design Studio programı açıldığında ekrana çıkan sayfada File -> New Project sekmesine tıklanır.



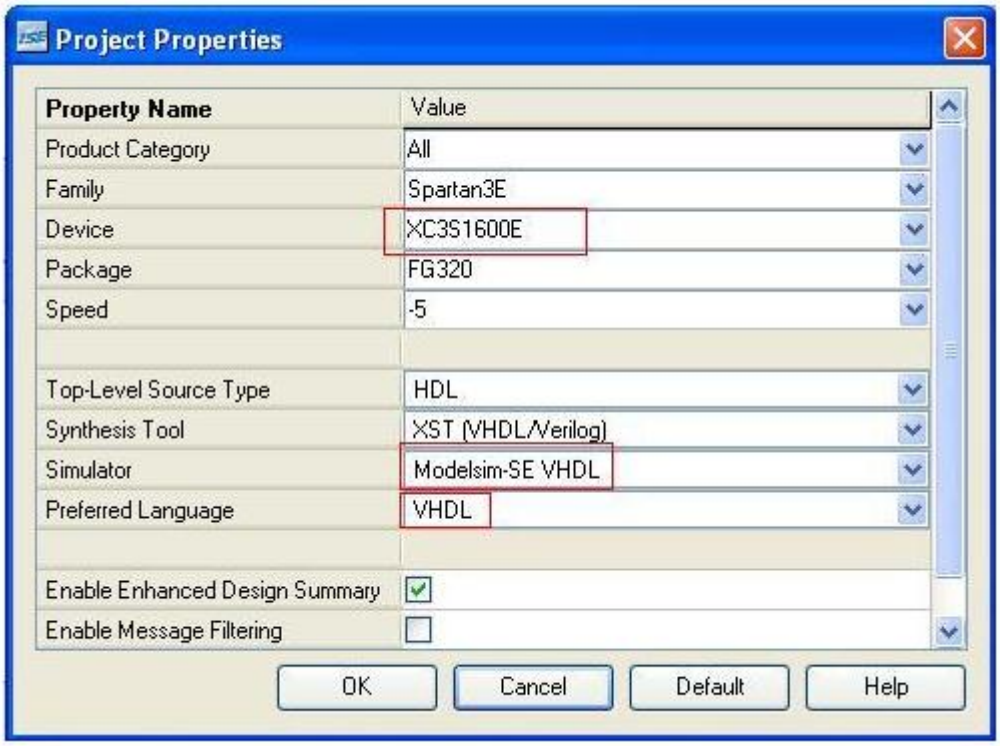
Şekil 3.9. ISE yeni proje oluşturma

Açılan pencereden Project Name kısmına proje ismini, Top Level source type kısmında HDL kısmını seçip Next' e basılır.



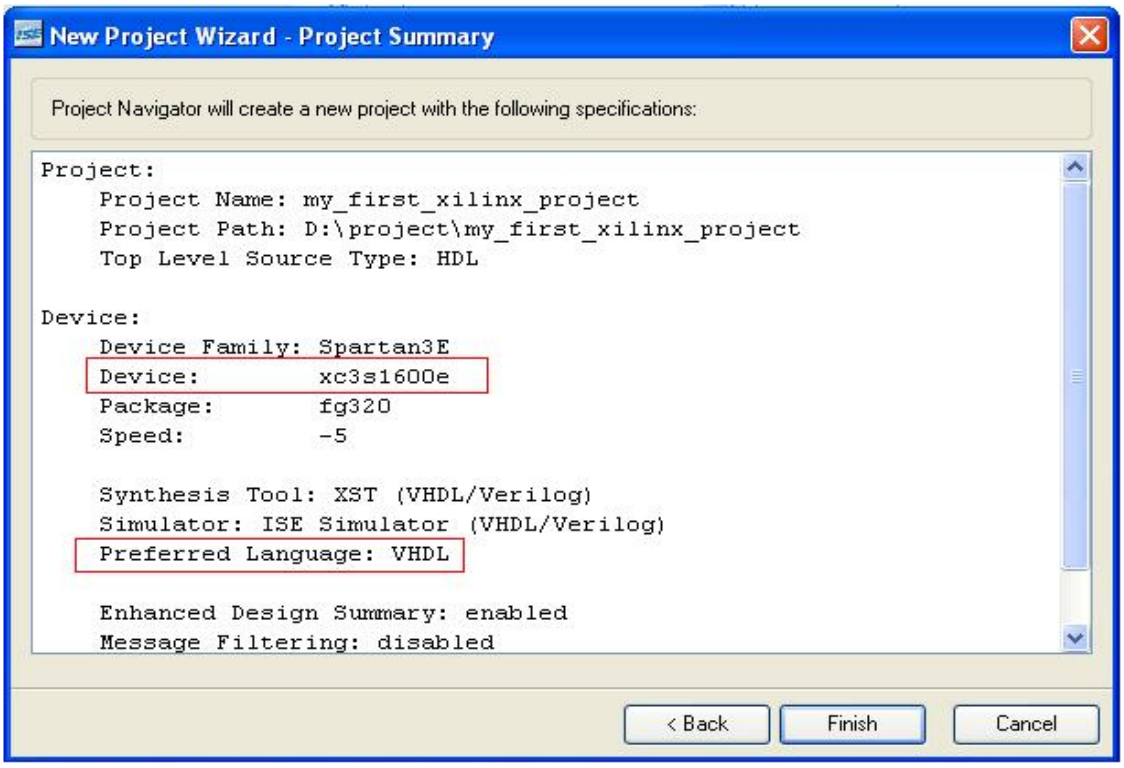
Şekil 3.10. Proje isimlendirme

Açılan yeni pencerede kullanılacak FPGA modeli ve preferred language kısmında kullanılacak donanım tanımlama dili seçilir.



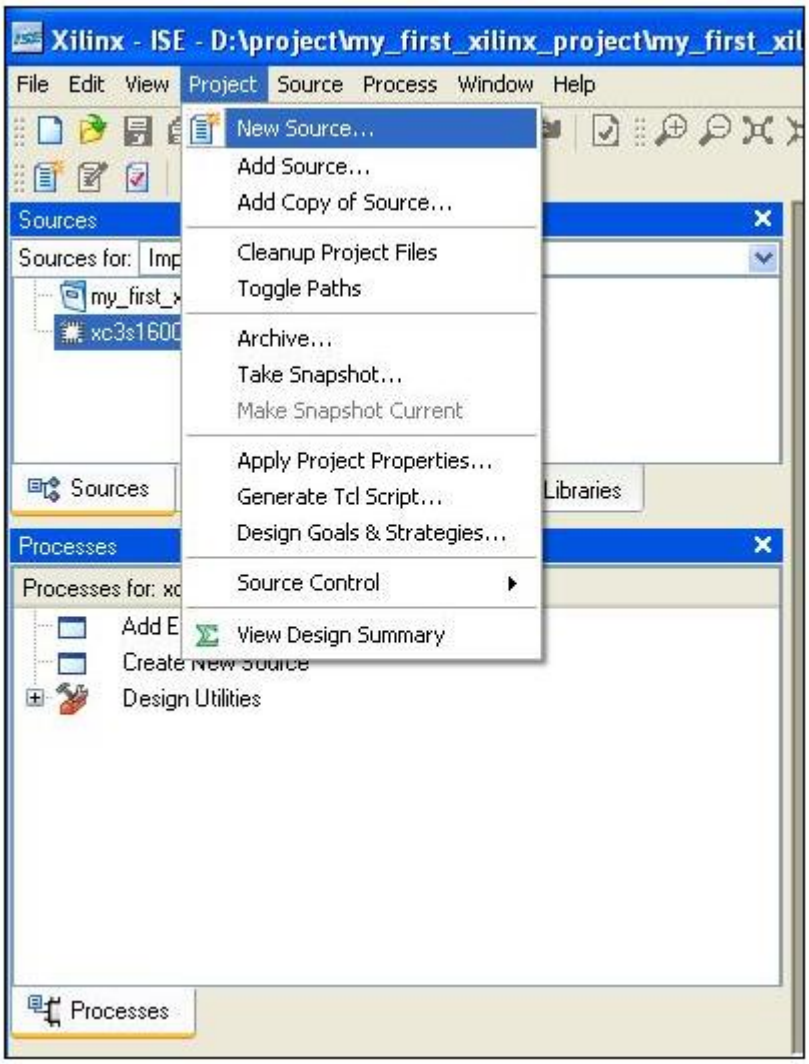
Şekil 3.11. Proje özelliklerinin seçilmesi

Açılan pencerede Finish sekmesine tıklanır.



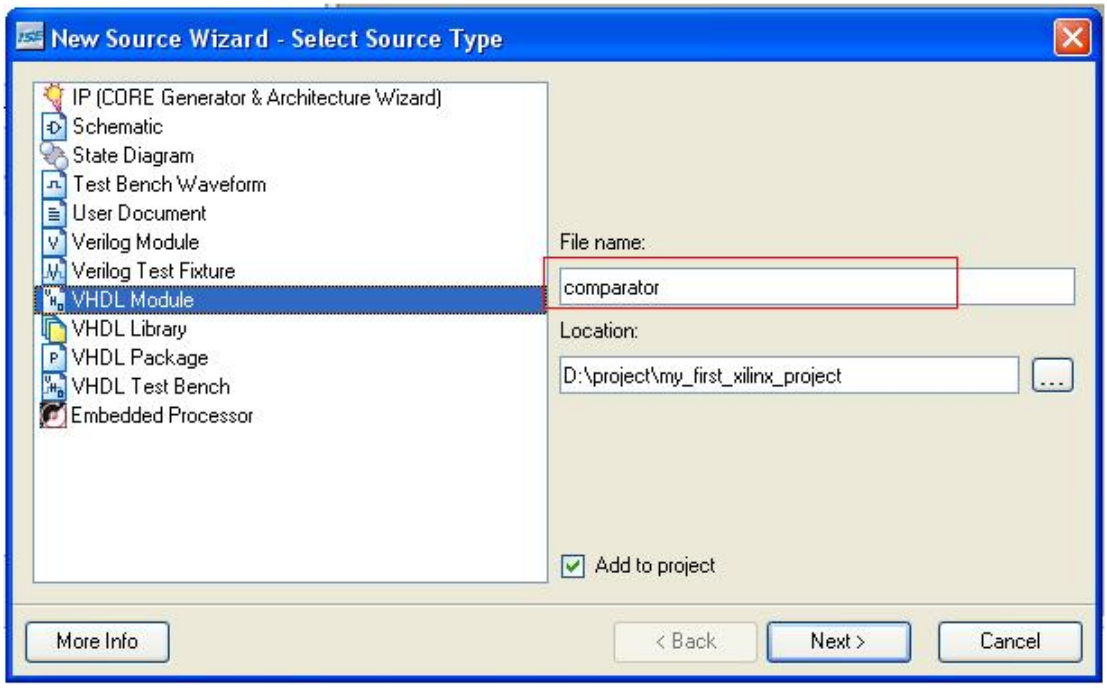
Şekil 3.12. Proje özetinin görüntülenmesi

Böylelikle yeni bir proje oluşturulmuş olur, bu işlemlerin ardından kodların yazılacağı klasörler oluşturulmalıdır. Project-> New Source penceresine tıklanır.



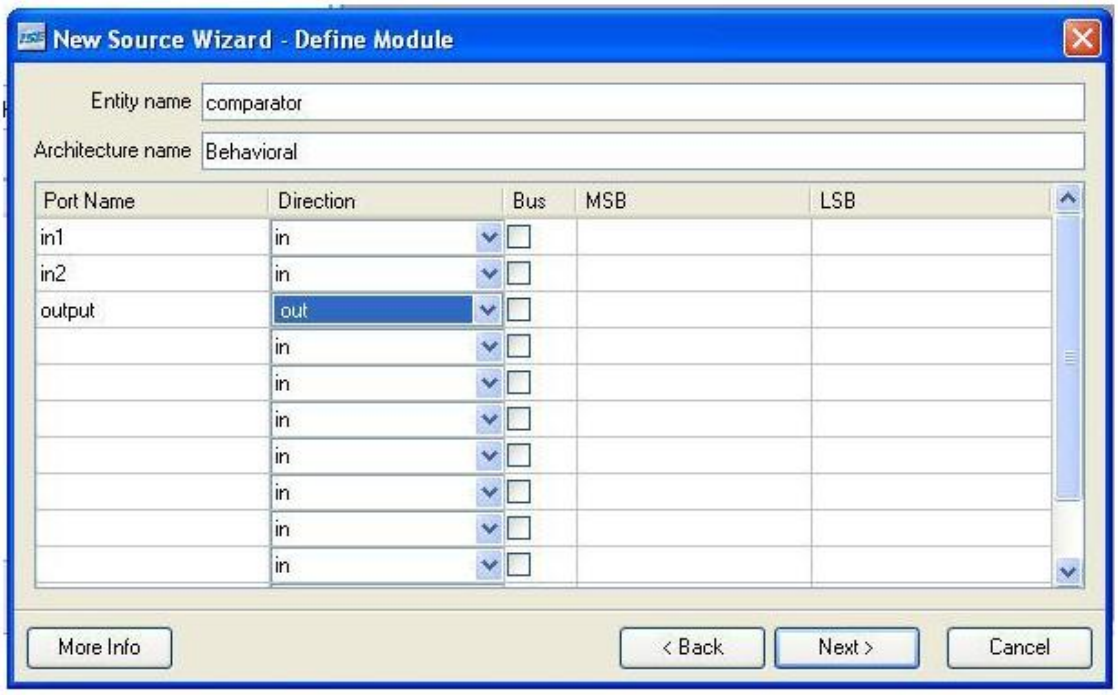
Şekil 3.13. Projeye VHDL modüle eklenmesi

Açılan pencerenin sol kısmında VHDL Module seçilerek File Name kısmına VHDL projesinin ismi girilerek Next'e tıklanır.



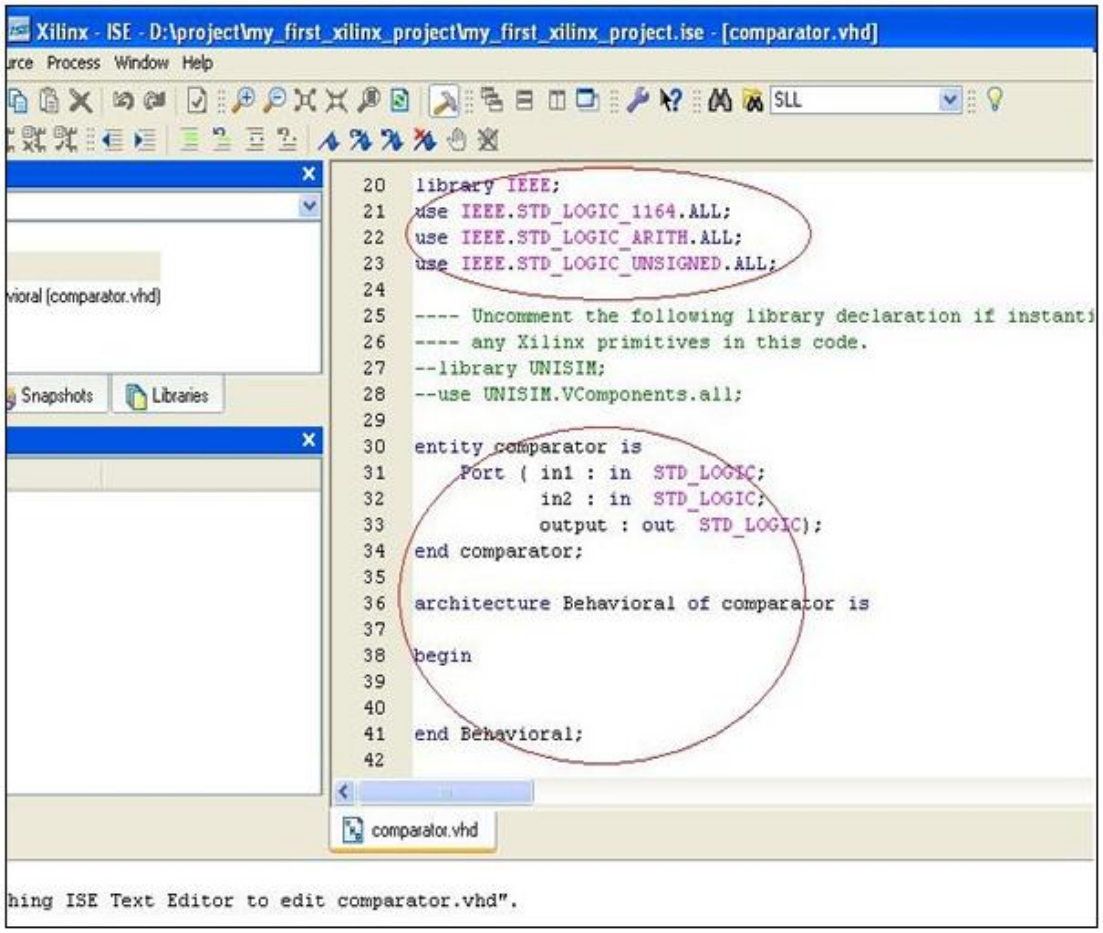
Şekil 3.14. VHDL modüle isimlendirme

Açılan pencerede Port Name ve Direction kısımlarına programda kullanılacak giriş/çıkış isim ve yönlendirmeleri seçilir.



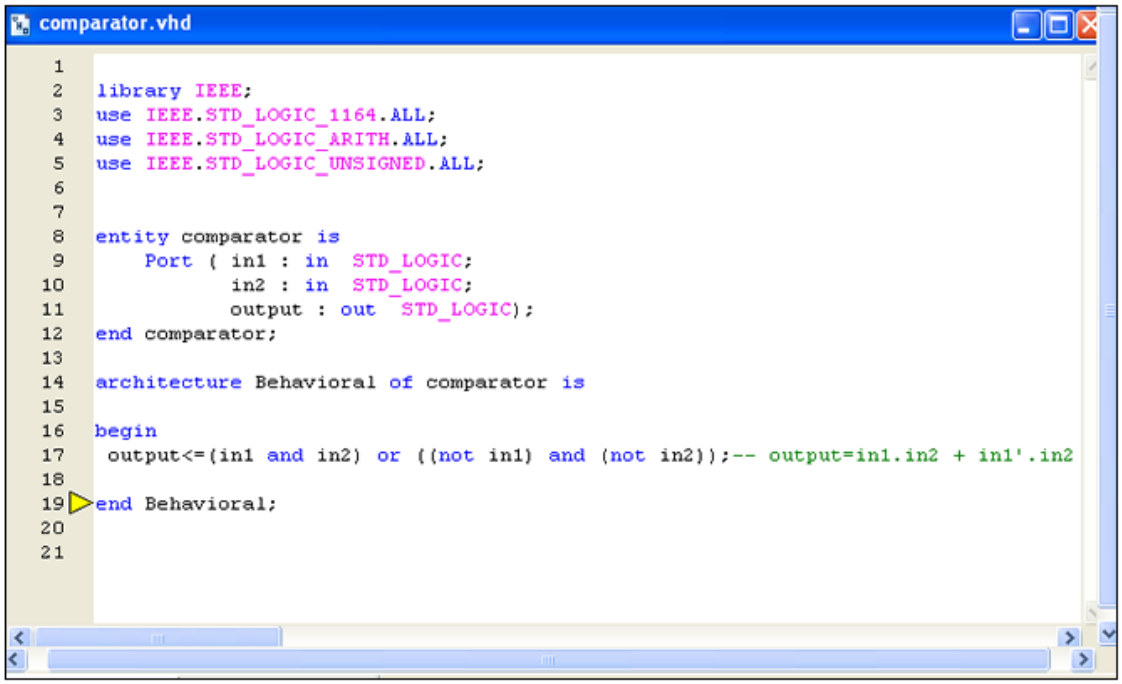
Şekil 3.15. VHDL modüle giriş çıkış port bilgileri

Açılan pencerede Finish sekmesine tıklanır, entity ve architecture kısımlarının otomatik olarak oluştuğu .vhd uzantılı dosya oluşturulur.



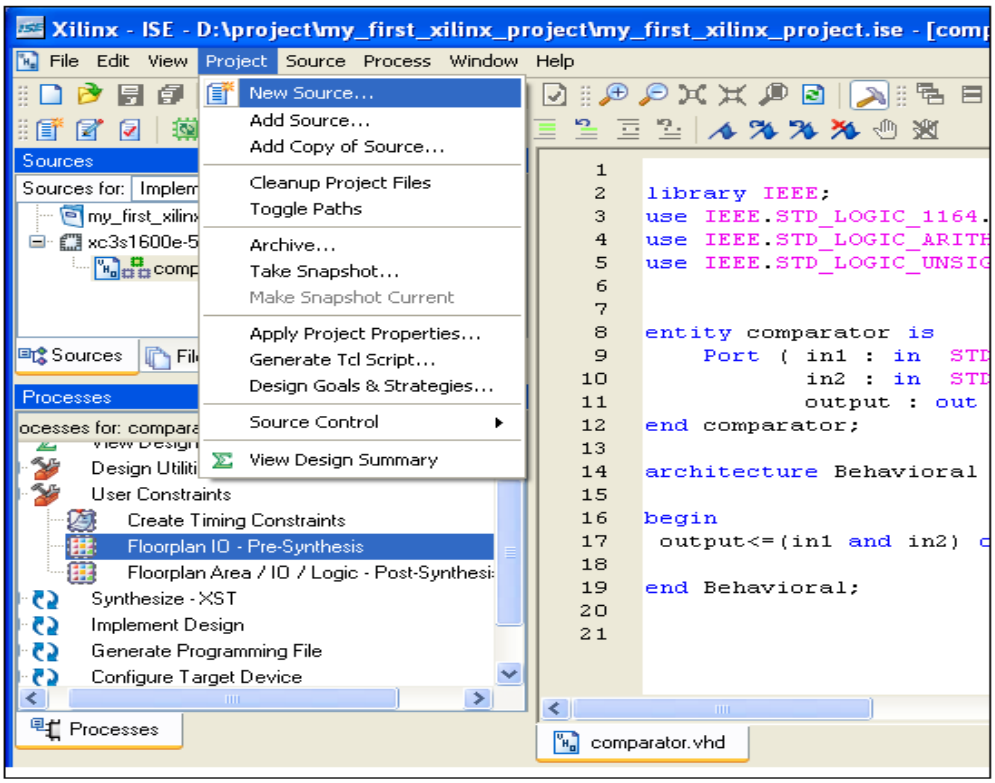
Şekil 3.16. VHDL modüle açılış sayfası

Oluşan .vhd dosyasının içeriği VHDL kodlarıyla tasarlanan program doğrultusunda yazılır.



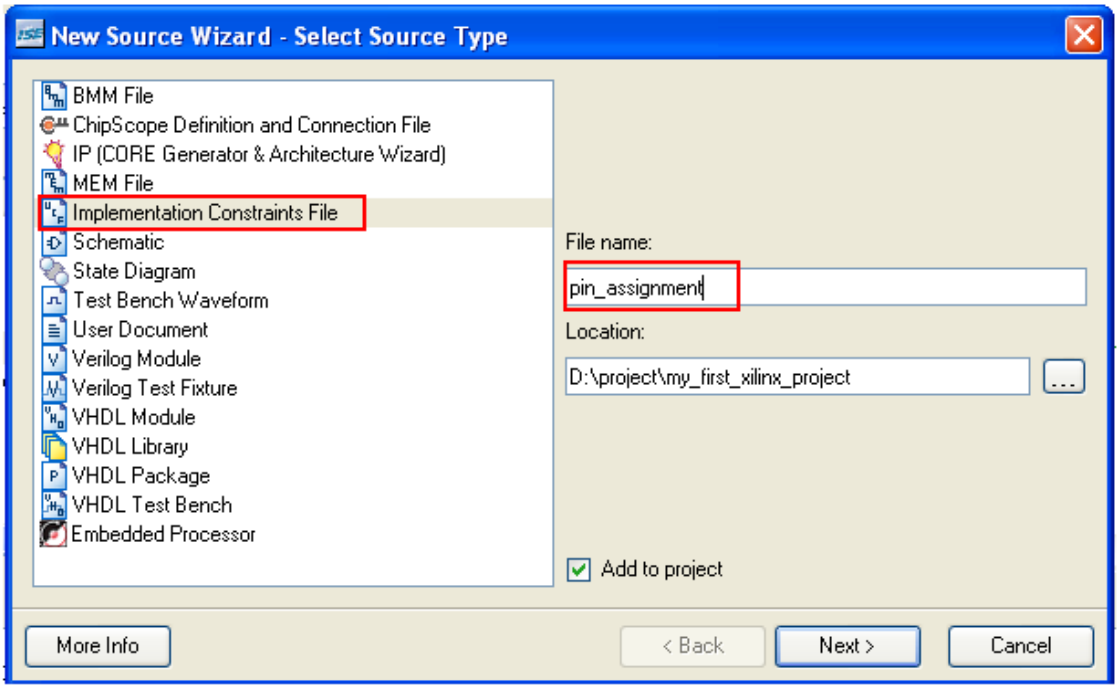
Şekil 3.17. Örnek VHDL kodu

Kod tamamlandıktan sonra kullanılacak FPGA pinlerine sinyaller atanmalıdır, pin ataması için .ucf uzantılı dosya oluşturulur.



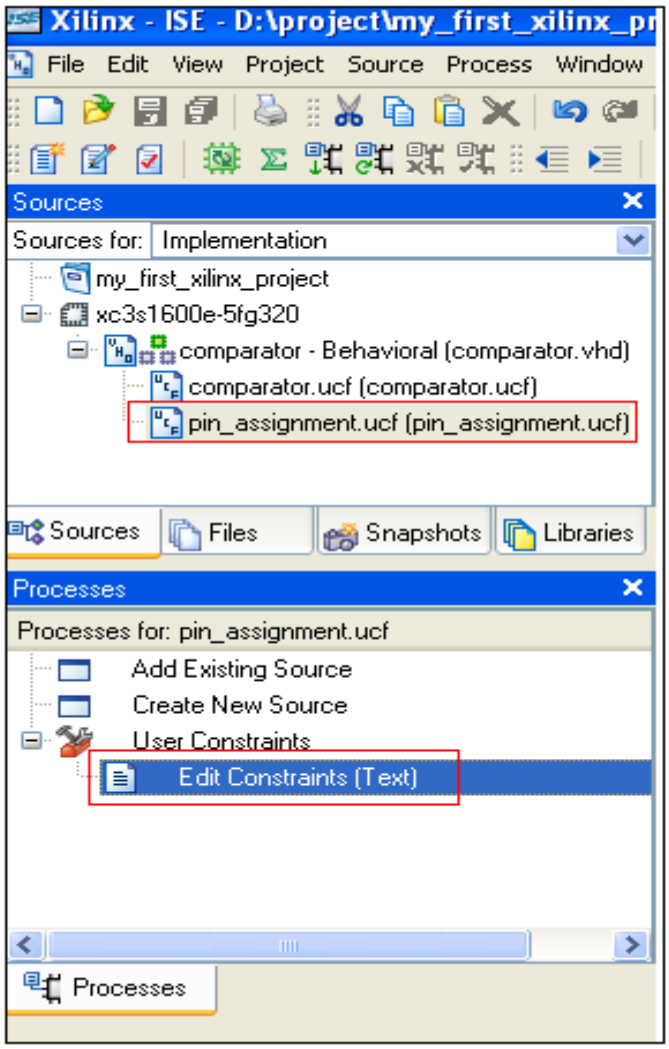
Şekil 3.18. Ucf dosyası ekleme

Proje -> New Source sekmesinde açılan pencerenin sol kısmından Implementation Constrait File seçilerek File_Name kısmına pin dosyasına verilerek isim girilir.



Şekil 3.19. ucf dosyası isimlendirme

Ardından Next' e tıklanır, processes kısmından Edit Constraints dosyasına çift tıklanır ve ekli sayfaya pin atama kodları yazılır.



Şekil 3.20. ucf dosyasının proje sayfasında görünümü

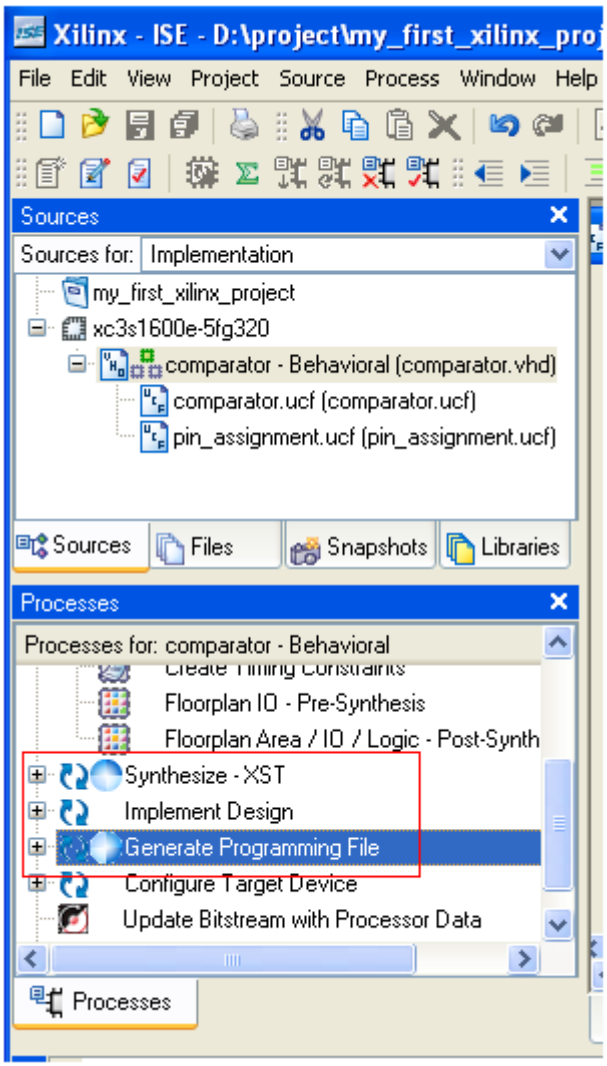
```
1  # ==== Clock inputs (CLK) ====
2  NET "CLK_50MHZ" LOC = "C9"  | IOSTANDARD = LVCMOS33 ;
3  #      Define clock period for 50 MHz oscillator (40%/60% duty-cycle)
4  NET "CLK_50MHZ" PERIOD = 20.0ns HIGH 40%;
5
6  # ==== Character LCD (LCD) ====
7  NET "LCD_E"  LOC = "M18" | IOSTANDARD = LVCMOS33  | DRIVE = 4  | SLEW = SLOW ;
8  NET "LCD_RS" LOC = "L18" | IOSTANDARD = LVCMOS33  | DRIVE = 4  | SLEW = SLOW ;
9  NET "LCD_RW" LOC = "L17" | IOSTANDARD = LVCMOS33  | DRIVE = 4  | SLEW = SLOW ;
10 # LCD data connections are shared with StrataFlash connections SF_D<11:8>
11 NET "SF_D<8>"   LOC = "R15" | IOSTANDARD = LVCMOS33  | DRIVE = 4  | SLEW = SLOW ;
12 NET "SF_D<9>"   LOC = "R16" | IOSTANDARD = LVCMOS33  | DRIVE = 4  | SLEW = SLOW ;
13 NET "SF_D<10>"  LOC = "P17" | IOSTANDARD = LVCMOS33  | DRIVE = 4  | SLEW = SLOW ;
14 NET "SF_D<11>"  LOC = "M15" | IOSTANDARD = LVCMOS33  | DRIVE = 4  | SLEW = SLOW ;
```

Şekil 3.21. Örnek ucf dosyası kodu

Bu yazım dili VHDL ‘ den farklı özel bir dildir. ’#‘ ile başlayan kısımlar açıklama satırlarıdır. Yani program sentezleme işlemi yaparken ‘#’ ile başlayan yerleri dikkate

almaz. Şekilde görüldüğü gibi tüm port tanımlamaları “NET” ile başlamaktadır. “NET” den sonra çift tırnak arasında port ismi bulunmaktadır. Burada *.vhd uzantılı dosyadaki portlara verilen isimlerin *.ucf uzantılı dosyada da aynı olması gerektiğine dikkat edilmelidir. “LOC” dan sonraki çift tırnak içindeki kısım en önemli kısımdır. Tanımlanan portun FPGA yongasının hangi pinine bağlanacağını belirler. Şekilde verilen örnek Spartan 3E Starter in kullanım kılavuzundan alınmıştır. Donanımsal olarak CLK C9 pininden verildiği için çift tırnak içerisine C9 yazılmıştır. “IOSTANDARD” ise portun tipini belirlemektedir. Verilen örnekte CLK “LVCMOS33” standartındadır. Satır 4’e tanımlanan kısım Spartan 3E Starter’a özgü satırdır. Değişik geliştirme kartlarında kullanılan clock tipine göre değişmektedir. “DRIVE” ve “SLEW” ise portun diğer fiziksel özellikleridir. FPGA pininin bağlandığı donanıma bağlıdır.

Oluşturulan iki dosyada da (*.vhd, *.ucf) gerekli değişiklik ve eklemeler yapıldıktan sonra kaydedilir. “Hierrarchy” kısmındaki “Behavioral…”in üzerine tıklanılır. Alt penceredeki listeden “Generate Programming File” e sağ tıklanır ve “Rerun All” tıklanır ve sentezleme işlemine başlanır. Yazılan kodlara bağlı olarak sentezleme süresi değişir. “Console” penceresinde sentezleme süreci hakkında bilgi verilmektedir. Eğer sentezleme işleminde hata bulunursa, işlem durur ve bu pencerede bulunan hatalar gösterilir. Bu hatalar giderilerek tekrar sentezleme başlatılır.



Şekil 3.22. Projenin derlenmesi

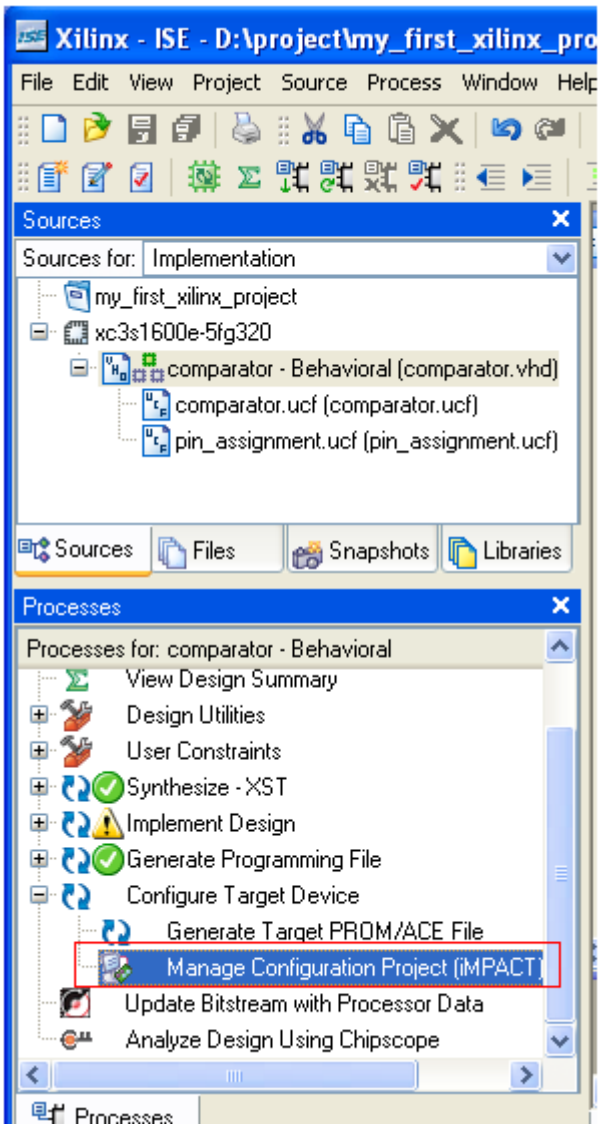
Eğer verilen hata mesajı *.ucf kaynaklı NET ile yapılan tanımlamalar ile ilgiliyse hata düzeltildiği halde sentezleme tekrar başlatıldığında aynı hata mesajı alınabilir. Bu kullanılan sentezleyicideki bir hatadır. Böyle bir durumda yeni proje eklenip hatalı projedeki kaynak dosyaları yenisine dahil edilerek tekrar sentezlenir.

Sentezleme işlemi bittikten sonra projenin oluşturulduğu klasörde *.bit uzantılı bir dosya oluşur. Bu dosyada oluşturulan donanımla ilgili her türlü bilgi mevcuttur.

3.5.1. Programın FPGA' ya gömülmesi

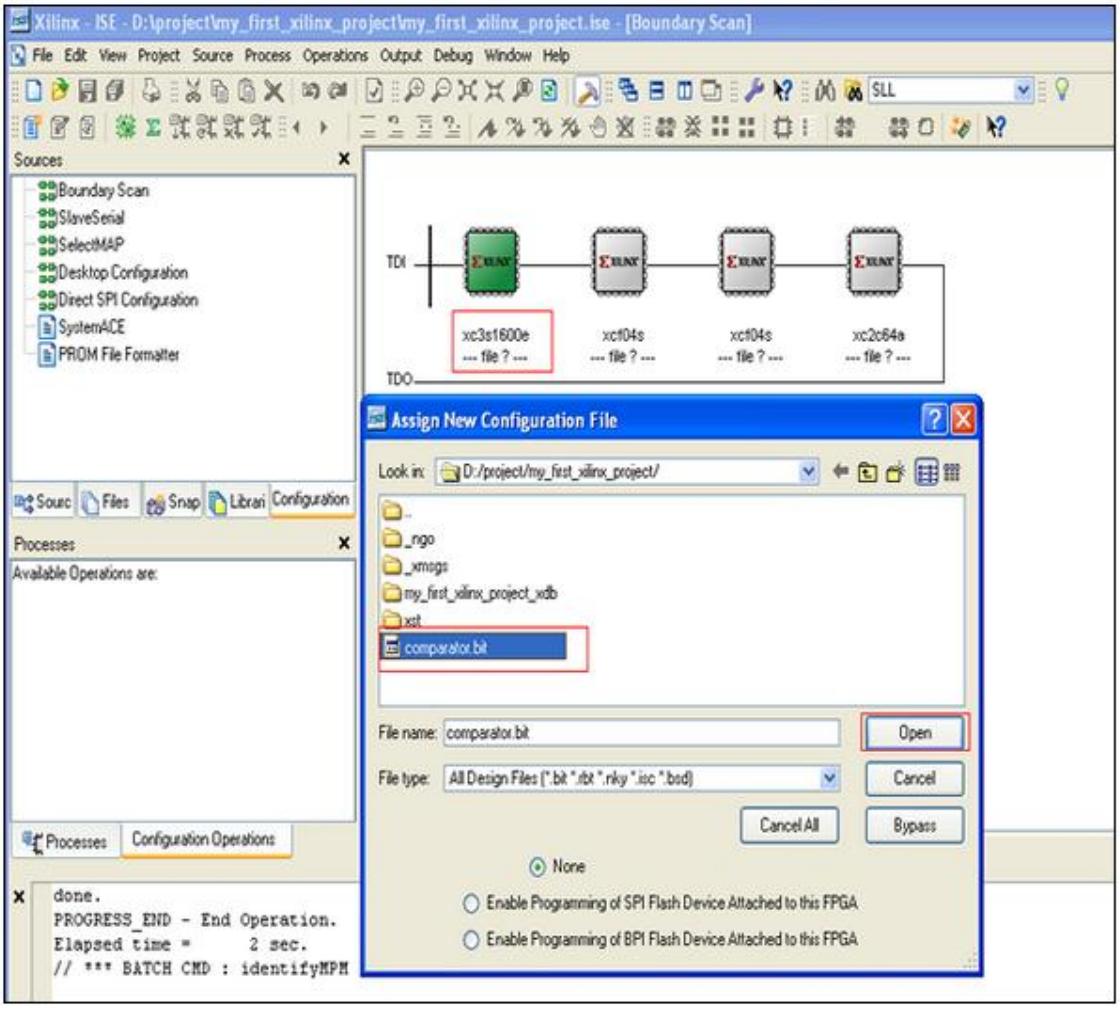
FPGA yongasına oluşturulan donanımın yüklenmesi geliştirme kartının özelliklerine göre farklılık gösterir. Spartan 3E Starter kartında sadece USB JTAG ile yükleme özelliği bulunurken, Spartan 3A DSP ve Virtex 5 kartlarında hem USB JTAG hem de Compact Flash ile yükleme özelliği bulunmaktadır. USB JTAG .bit uzantılı dosyayı doğrudan karta yollayarak FPGA yongasını boot etmektedir. Bu şekilde boot edildiğinde kartın kapatılıp açıldığında FPGA yongası hiç programlanmamış gibi olur. Bu sayede tekrar tekrar programlanabilmektedir. Diğer bir yöntem ise .bit uzantılı dosyayı geliştirme kartı üzerindeki System ACE nin okuyacağı .ace uzantılı formata çevirerek Compact Flash' a yüklemektir. Bu yükleme şekliyle geliştirme kartı çalıştırılınca System ACE, Compact Flash 'tan .ace uzantılı dosyayı okur ve FPGA yongasını boot eder. Bu sayede yonganın programlanan şekle gelmesi boot süresi kadar kısa bir sürede tamamlanır.

Oluşturulan .bit uzantılı dosyanın FPGA üzerine yüklenebilmesi için IMPACT arayüzü kullanılır. Configure Target Device'ın altında bulunan Manage Configuration Project(IMPACT) üzerine tıklanarak bu arayüze ulaşılır.



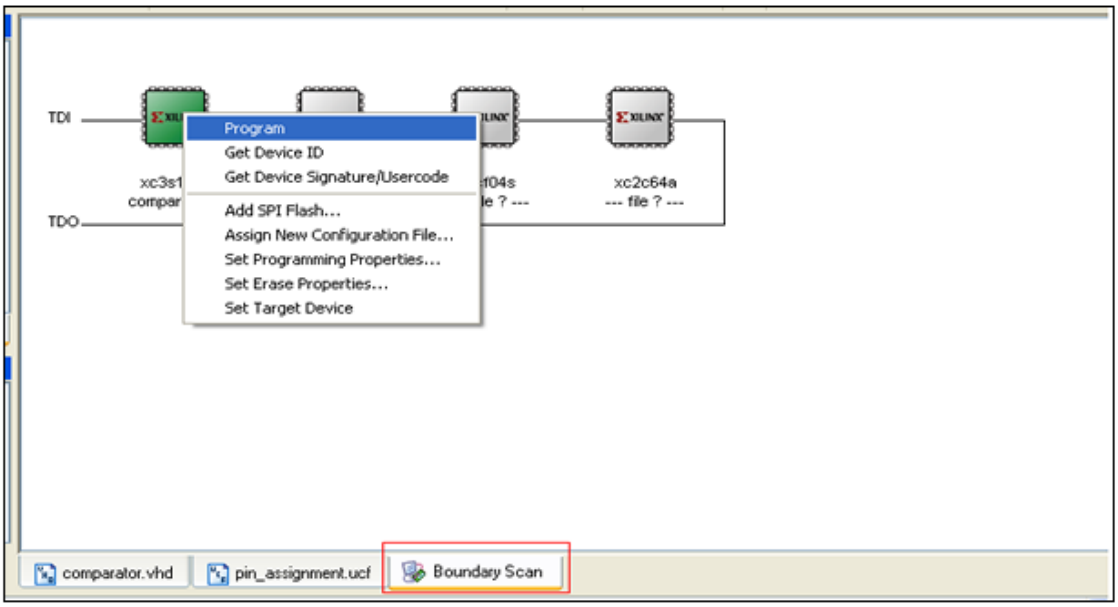
Şekil 3.23. IMPACT ara yüzüne erişim

Açılan Boundary Scan penceresinde FPGA' e yüklenecek .bit uzantılı dosya seçilerek Open' a basılır.

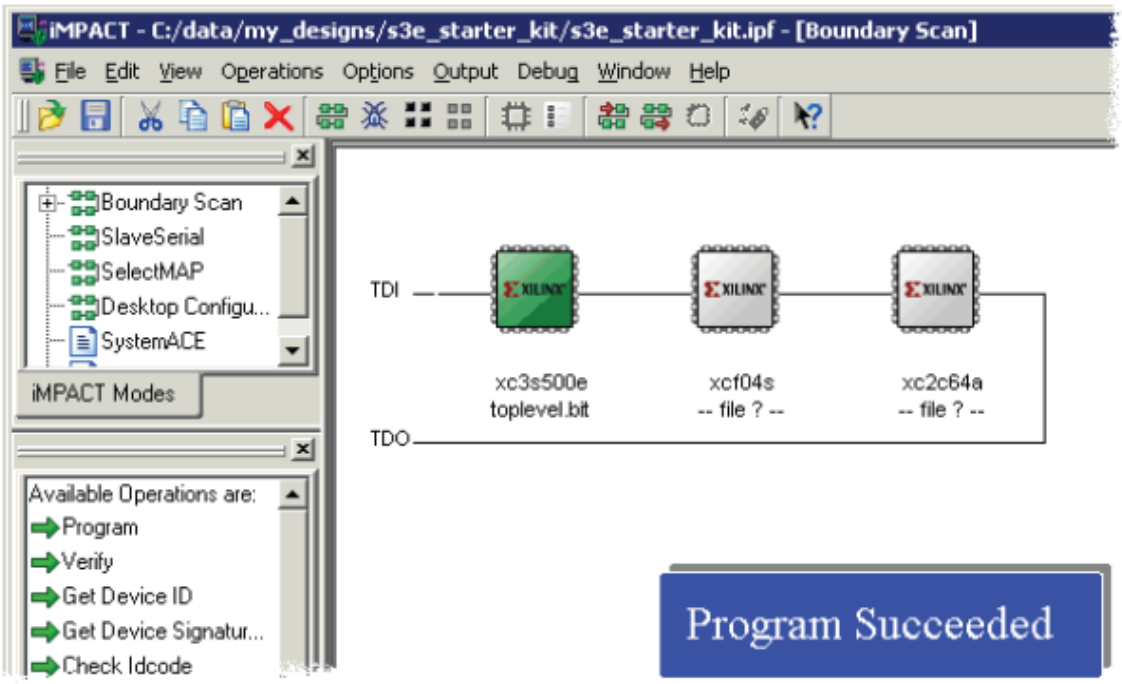


Şekil 3.24. bit uzantılı dosyanın FPGA' ya aktarımı

Ardından FPGA sembolünün üzerine gelip sağ tıklayarak Program seçilir, Program Succeeded ibaresi göründüğünde tasarlanan program FPGA üzerine başarıyla yüklenmiş olur.



Şekil 3.25. Programın FPGA'ya yüklenmesi

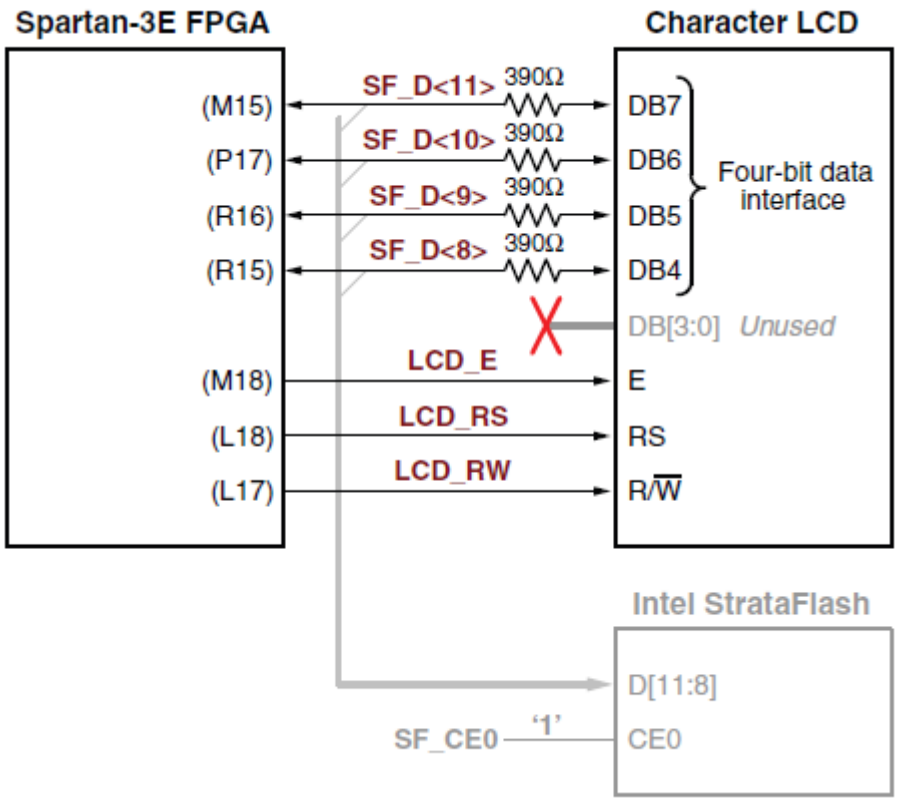


Şekil 3.26. Programın doğru şekilde yüklenmesi

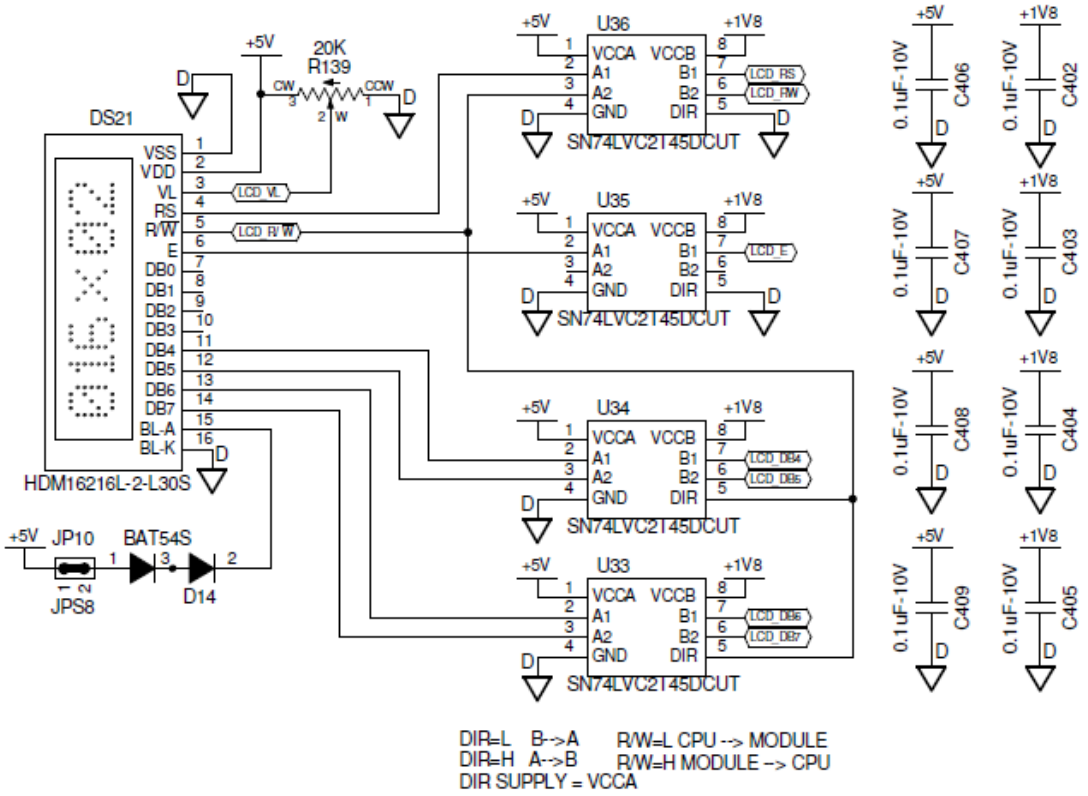
4. FPGA ÜZERİNDE STEGANOGRAFİ UYGULAMASI

Tez çalışmasında FPGA üzerinde steganografi uygulaması gerçekleştirmek hedeflenmiştir. Spartan 3E geliştirme kartı üzerindeki 2*16 satır LCD ekran üzerinde metin gizleme uygulaması yapılmıştır.

4.1. Spartan 3E Geliştirme Kartı Üzerindeki LCD Ekranı Sürme



Şekil 4.1. Spartan 3E FPGA ile LCD arası bağlantı (User Guide 2011)



Şekil 4.2. LCD şematik bağlantıları (User Guide 2011)

Şekil 4.1‘ de gösterilen bağlantılardan DB0-DB7 pinleri data biti olarak kullanılmaktadır. Tez çalışmasında kullanılan FPGA, LVTTL seviyesinde çalıştığından maksimum 3,3 V‘ luk gerilime izin vermektedir. Ancak kullanılan LCD 5V gerilim seviyesiyle çalıştığından sırayla gösterilen dört eleman 5V’ u FPGA gerilime uygun olarak 1,8 V seviyesine dönüştürmek için kullanılmaktadır.

LCD ‘yi kullanabilmek için ayarlanması gereken 7 pin mevcuttur. Bunlar;

- SF_D<11> data biti DB7
- SF_D<10> data biti DB6
- SF_D <9> data biti DB5
- SF_D <8> data biti DB4
- LCD_E okuma / yazma izin pini (0: izin vermez, 1: okuma/yazma izni)
- LCD_RS saklayıcı seçim pini

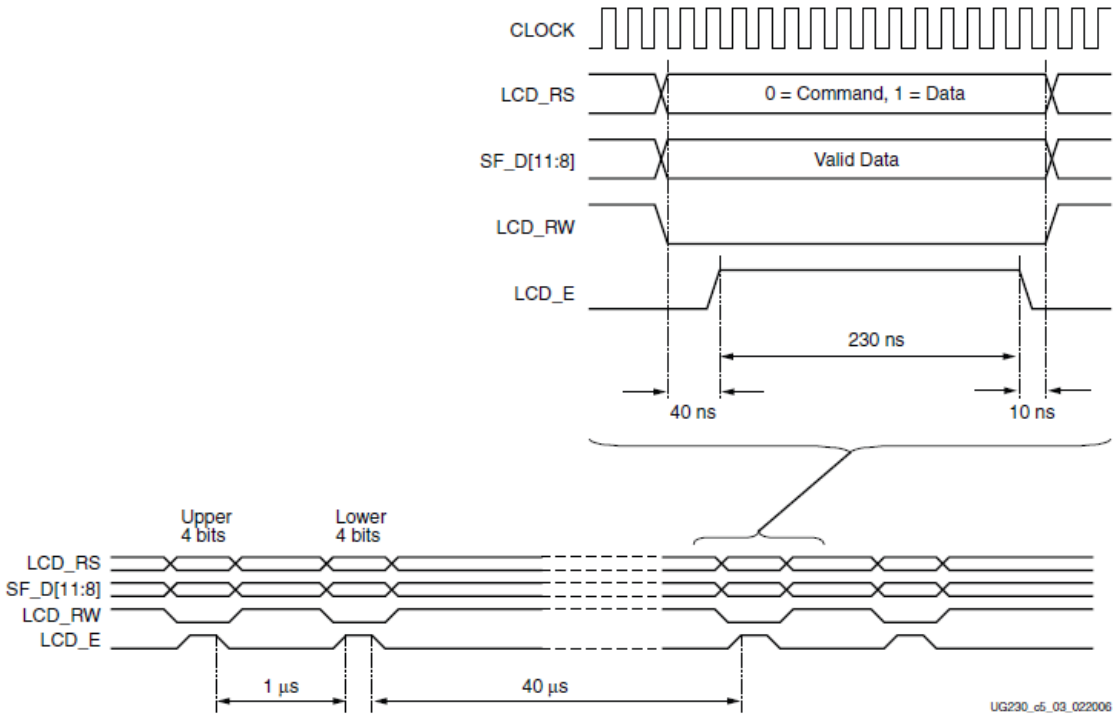
(yazma ve okuma işlemleri boyunca veri gönderebilmek için lojik 1 konumuna ayarlanmalıdır.)

- LCD_RW okuma/yazma kontrol pini
  (0:yazma, LCD verileri kabul eder 1: okuma, LCD verileri gösterir)

LCD ‘ yi aktif hale getirmek ve 4 bitlik uygulama oluşturabilmek için aşağıdaki basamaklar sırasıyla yerine getirilir.

1. FPGA ‘ nın konfigürasyon ayarlarını sonlandırıp uygulamaya başlayabilmesi için yaklaşık 15 ms’ lik bir süreye ihtiyaç vardır. Bu süre 50 MHz çalışma frekansında 750.000 clock darbesi üretilerek sağlanabilir.
2. SF_D<11:8> veri gönderme pinlerine 0*3 yazılır. Bu işlem LCD_E pini 12 clock darbesi boyunca lojik 1 seviyesinde konumlandırılmasıyla gerçekleştirilir.
3. 4.1 ms beklenir bu işlem 50 MHz çalışma frekansında 205.000 clock darbesi üretilmesiyle sağlanır.
4. SF_D<11:8> veri gönderme pinlerine 0*3 yazılır. Bu işlem LCD_E pini 12 clock darbesi boyunca lojik 1 seviyesinde konumlandırılmasıyla gerçekleştirilir.
5. 100 us beklenir, bu işlem 50 MHz frekansta 5000 clock darbesi üretilerek gerçekleştirilir.
6. SF_D<11:8> veri gönderme pinlerine 0*3 yazılır. Bu işlem LCD_E pini 12 clock darbesi boyunca lojik 1 seviyesinde konumlandırılmasıyla gerçekleştirilir.
7. 40 us beklenir, bu işlem için 50 MHZ de 2000 clock darbesi üretmeye ihtiyaç vardır.
8. SF_D<11:8> veri gönderme pinlerine 0*2 yazılır. Bu işlem LCD_E pini 12 clock darbesi boyunca lojik 1 seviyesinde konumlandırılmasıyla gerçekleştirilir.
9. 40 us beklenir, bu işlem için 50 MHZ de 2000 clock darbesi üretmeye ihtiyaç vardır.

4 biti değerli 4 biti değersiz olmak üzere toplam 8 bitlik veri gönderme işlemi için kullanılması gereken zamanlama ilişkileri ve clock darbe süreleri aşağıdaki şekilde gösterilmiştir.



Şekil 4.3. LCD veri yazma protokolü (User Guide 2011)

LCD kullanmak için gerekli olacak komut satırlarıyla ilgili veriler aşağıdaki çizelgede yer almaktadır.

Çizelge 4.1. LCD bitlerinin durumuna göre modlar (User Guide 2011)

Function	LCD_RS	LCD_RW	Upper Nibble				Lower Nibble			
			DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Clear Display	0	0	0	0	0	0	0	0	0	1
Return Cursor Home	0	0	0	0	0	0	0	0	1	-
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S
Display On/Off	0	0	0	0	0	0	1	D	C	B
Cursor and Display Shift	0	0	0	0	0	1	S/C	R/L	-	-

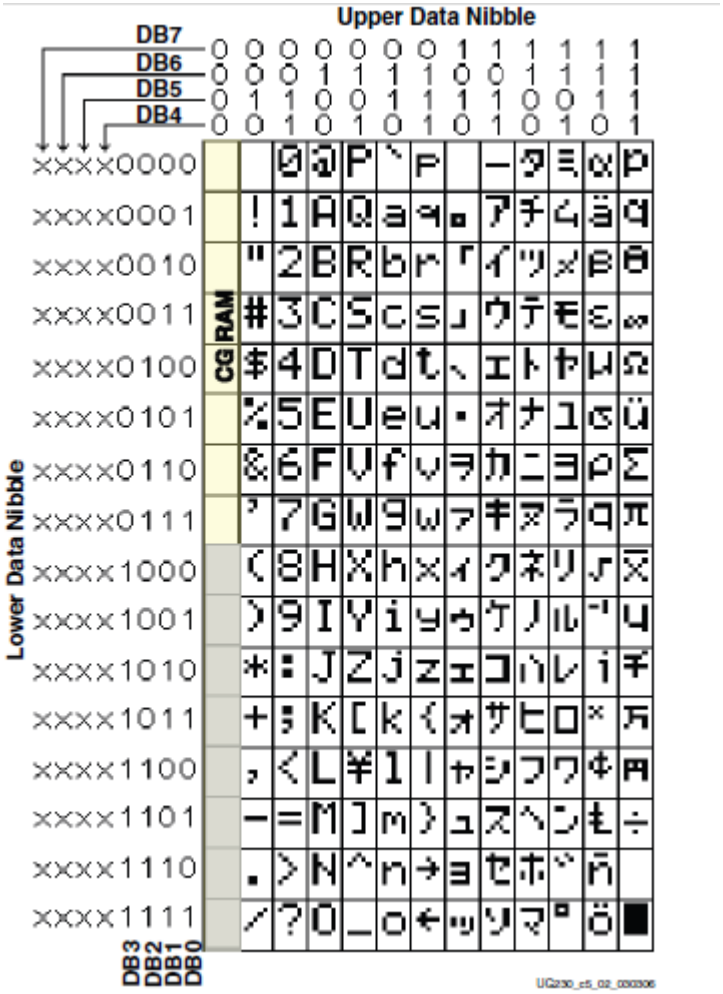
Yukarıda dokuz madde halinde belirtilen LCD yi aktif kılma işlemleri sırasıyla gerçekleştirildikten sonra 4’ er bit halinde uygulama yapılması için yapılması gereken adımlar aşağıda yer almaktadır;

- Function set komutu çalıştırılarak 0*28 verisi gönderilir.
- Entry Mode Set komutu çalıştırılarak 0*06 verisi gönderilir.

- Display on/off komutu çalıştırılarak 0*0C verisi gönderilerek kursörün yanıp sönmesi sağlanır.
- Son olarak Clear Display komutu çalıştırılarak 82.000 clock darbesi kadar beklenilir.

Bütün veri gönderme işlemlerinde Şekil ‘deki zamanlama işaretleri referans alınmalıdır. Gönderilebilecek karakterler aşağıda verilmiştir.

Çizelge 4.2. LCD Karakter Seti (User Guide 2011)



4.2.FPGA Üzerinden VHDL Donanım Tanımlama Dili LCD Ekran Sürme

Belirtilen LCD özellikleri ve işletilmesi gereken adımlar için kullanılabilecek VHDL kod kısmı aşağıdaki şekilde başlatılabilir.

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_ARITH.ALL;
4  use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6
7
8  entity lcddeneme is
9      Port ( clk : inout  STD_LOGIC;   --- giriş ve çıkış pinlerinin tanımlanması
10            lcd_rs : out  STD_LOGIC;
11            lcd_e : out  STD_LOGIC;
12            lcd_rw : out  STD_LOGIC;
13            dat7 : out  STD_LOGIC;
14            dat6 : out  STD_LOGIC;
15            dat5 : out  STD_LOGIC;
16            dat4 : out  STD_LOGIC);
17 end lcddeneme;
18
19 architecture Behavioral of lcddeneme is
20
21 begin
22
23 lcd_rw <= '0';                        -- yazma bitinin lojik 1 e ayarlanması
24
25    process(clk)                        -- clock 'a bagli processin tanımlanması
26
27    variable cnt : integer range 0 to 999999; --süre belirleme sayacının tanımlanması
```

```
28    begin
29    if rising_edge(clk) then
30       cnt :=cnt +1;                 -- 0 başlancına sahip sayacın saymaya 1 den başlaması
31       if (cnt = 1) then
32
33       lcd_e <= '0';
34       elsif (cnt = ******) then     --0x3 verisi gönderilir.
35
36       dat7 <= '0';
37       dat6 <= '0';                  --3
38       dat5 <= '1';
39       dat4 <= '1';
40       elsif (cnt = ******) then
41       lcd_e <= '1';
42       elsif (cnt = ******) then
43       lcd_e <= '0';
44       elsif (cnt = ******) then
45
46       dat7 <= '0';
47       dat6 <= '0';                  --3
48       dat5 <= '1';
49       dat4 <= '1';
50
51       lcd_e <= '1';
52       elsif (cnt = ******) then
53       lcd_e <= '0';
54       elsif (cnt = ******) then
```

Şekil 4.4. LCD sürmek için örnek VHDL kod başlangıcı

“ ******” ile gösterilen kısımlar çalışılan frekans değerine göre hesaplanması gereken clock darbelerinin değerleridir. Hesaplanma şekli aşağıda belirtilmiştir.

50 MHz lik bir sistem frekansı için 40 ns lik bir süre

1/50 MHz = 20 ns(sistem periyodu) 40ns/20 ns= 2 clock darbesi biçiminde hesaplanır. Hesaplanan uygun değerler yerlerine yazılarak kodun devamı getirilir.

VHDL kodunun yazılmasından sonra ucf dosyası Spartan 3E için aşağıdaki şekilde düzenlenmelidir.

```
1
2
3  NET "clk" LOC = "C9" | IOSTANDARD = LVCMOS33 ;
4  NET "clk" PERIOD = 20.0ns HIGH 40%;
5
6  NET "lcd_e" LOC = "M18" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
7  NET "lcd_rs" LOC = "L18" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
8  NET "lcd_rw" LOC = "L17" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
9  NET "dat7" LOC = "M15" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
10 NET "dat6" LOC = "P17" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
11 NET "dat5" LOC = "R16" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
12 NET "dat4" LOC = "R15" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
```

Şekil 4.5. LCD için ucf dosyası

4.3. LCD Üzerinde Steganografi Algoritması

LCD üzerinde veri gizleme uygulaması, metin içine metin gizleme yöntemi yöntemiyle gerçekleştirilmiştir. Literatürde de örnekleri bulunan orijinal mesajın boşlukları arasına gizli mesaj gizlenerek metin steganografi uygulaması hedeflenmiştir (Singh vd., 2012).

Gönderici Tarafı Algoritma Basamakları:

- LCD ekrana gönderilecek kelimeler arası verilen boşluklara sırasıyla gizlenmesi istenen kelimenin harfleri gönderilecektir.
- Gizlenecek kelimenin harfleri sırasıyla hex değerine karşılık gelen binary formata çevrilir.
- Çevrilen binary formatındaki 1 ve 0 değerlerinin tersi alınarak yeni bir sekiz bitlik karakter elde edilir.
- Bu karakter LCD de görünmesini istediğimiz kelimelerin arasında bırakılan boşlukların yerine gönderilir.
- Ancak LCD ekran hex 7F karakterinden büyük değerleri ekranda göstermediğinden gönderilen gizli veri LCD ekranda boşluk olarak görünür.

- Kelimeler arası bırakılan tüm boşluklarda gizlenmek istenen veri bu algoritma çerçevesinde LCD' ye gönderilerek gizli veri aslında ekrana ulaştırılmış olur.

Alıcı Tarafı Algoritma Basamakları:

- LCD Ekranda kelimeler arası boşluk olarak görünen yerlerde gizlenmiş veri olduğu bilinmektedir.
- LCD ye veri gönderen yazılımda boşluğa denk gelen bölümler bulunur.
- Gönderilmiş olan binary karakterin tersi alınarak yeni bir sekiz bit elde edilir.
- Bu sekizli hexadecimale çevrili karakter karşılığı bulunur.
- Tüm boşluklarda bulunan karakterler aynı algoritma ile geri elde edilerek gizli veri çözülmüş olur.

Şekil 4.6. Spartan 3E geliştirme kartı

Örnek 1 : orijinal mesaj : 'MSc Thesis' gizli mesaj : 'fpga'

Şekil 4.7. Gizli veriyi içeren ekran görüntüsü

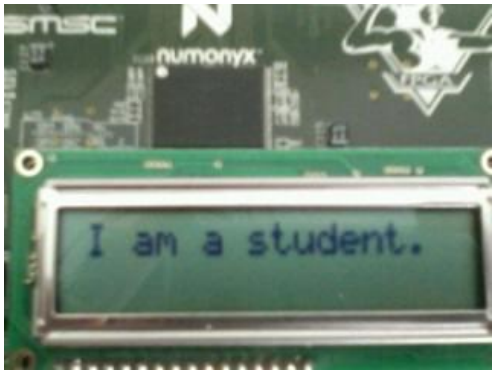
Şekil 4.8. Gizli verinin çözülerek ekrana gönderilmesi

Örnek 2: orijinal mesaj : ‘My final work ’ gizli mesaj :’ VHDL’

Şekil 4.9. Gizli veriyi içeren ekran görüntüsü

Şekil 4.10. Gizli verinin çözülerek ekrana gönderilmesi

Örnek 3: orijinal mesaj : ‘I am a student.’ gizli mesaj :’IEEE’



Şekil 4.11. Gizli veriyi içeren ekran görüntüsü

Şekil 4.12. Gizli verinin çözülerek ekrana gönderilmesi

5. SONUÇ

Tez çalışmasında son yıllarda paralel veri işleme yeteneği ile kullanılmaya başlanan FPGA platformu ile veri gizleme yöntemlerinden steganografi dalında bir uygulama gerçekleştirmeye yönelik çalışılmıştır.

Öncelikle Spartan 3E geliştirme kartı özellikleri, şematik bağlantıları ve kılavuzları incelenerek cihazın programlama özellikleri anlaşılmıştır. Ardından cihazı programlayabilmek için donanım tanımlama dilleri incelenmiştir. En fazla kaynak ve dokümana sahip olan VHDL donanım tanımlama dili tez çalışmasında kullanılmak üzere belirlenmiştir. VHDL, Xilinx ISE Design Studio derleyici ve sentezleyici programda kod satırıyla yazılarak öğrenilmiş ve uygulama geliştirebilecek aşamaya gelinmiştir.

Bir sonraki aşamada, literatürde kullanılan veri gizleme teknikleri araştırılarak steganografi bilim dalı tez çalışması için seçilmiştir. Steganografinin kullanım yöntemlerine göre metin, ses ve görüntü steganografi alanları ve uygulamaları incelenmiştir. Sonrasında, FPGA ile entegre edilebilecek metin steganografi seçilmiştir.

Kart üzerinde bulunan 2*16 satır LCD ekran üzerinde metin steganografi uygulaması geliştirmek için kelime aralarındaki boşluklara karakter gizleme işlemi algoritma olarak belirlenmiş ve bu algoritma veri gönderici ve alıcı taraf olmak üzere çalıştırılmıştır.

Algoritma sırasıyla üç farklı orijinal mesaj ve gizli mesaj kelimeleri belirlenerek uygulanmıştır. Orijinal mesajların kelimeleri arasındaki boşluklara gizli mesajların harfleri sırasıyla gizlenerek FPGA üzerinden LCD ekrana gönderilmiştir. Orijinal ve gizli mesajların birlikte LCD ekranda görüntülenme sonuçlarına tez çalışması içinde yer verilmiştir. Ayrıca, gizli mesajların alıcı tarafında çözülerek LCD ekran görüntüleri de çalışmada mevcuttur.

Yüksek lisans tez çalışmasında hedeflenen FPGA üzerinde veri gizleme uygulamaları belirtilen materyal ve algoritmayla başarılı sonuçlar alınarak gerçekleştirilmiştir.

KAYNAKLAR

Anderson R.J.1996. ed., Information Hiding: First International Workshop, vol 1174 of Lecture Notes in Computer Science, Isaac Newton Institute, Cambridge, England, Springer-Verlag, Berlin, Germany, ISBN 3-540-61996.

Bender W., Gruhl D., Morimoto N., Lu A.1996.“Techniques for data hiding”, IBM Systems Journal, vol. 35, NOS 3&4.

Brassil J.1995.“Electronic Marking and Identification Techniques to Discourage Document Copying” IEEE Journal On Selected Areas In Communications.Vol.13.No.8,October 1995

Chaum D.1981. “Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms”, Communications of the ACM, vol. 24, no. 2, pp. 84-88.

Goldschlag D.M. 1996. Reed M.G., Syverson P.F., “Hiding Routing Information”, in Information Hiding: First International Workshop, Proceedings, vol 1174 of Lecture Notes in Computer Science, Springer, pp. 84-88.

Gruhl D., Lu A., Bender W.1996. “Echo Hiding”, in Proceeding of First International Workshop, Springer, Cambridge, UK.

Hartung F., Kutter M.1991. “Multimedia watermarking techniques,” Proc. of the IEEE, vol. 87, pp. 1079–1107.

Heredotus.1971. “The Histories”, First Published in Greek 430 B.C.E., Translated by Selincourt A., Penguin Classics, Hammondsworth.

Johnson N.F, Jajodia S.1998. “Exploring steganography: Seeing the Unseen”, Computer, 31, no 2:26-34.

Johnson N.F., Duric Z., Jajodia S.2000. “Information Hiding: Steganography and Watermarking - Attacks and Countermeasures”, Kluwer Academic Publishers, ISBN: 0-79237-204-2.

Johnson N.F., Rude T.2001. “Introduction to Steganography Hidden Information”, Regional Computer Forensic Group (RCFG) and Mid-Atlantic Chapter of High-Technology Crime Investigation Association (HTCIA) George Mason University Computer Forensics Symposium (GMU 2001), Fairfax, VA, August 13-17.

Kabetta H, Dwiandiyanta Y.2011. “Information Hiding in CSS: A Secure Scheme Text-Steganography Using Public Key Cryptosystem”, International Journal on Cryptography and Information Security(IJCIS),Vol.1, No.1,December 2011.

Kahn D.1967. “The Codebreakers”, Macmillan, New York.

Katzenbeisser S.2006. Petitcolas F.A.P., "Information Hiding Techniques for Steganography and Digital Watermarking", Artech House, INC. 685 Canton Street Norwood, MA 02062.

Kharrazi M., Sencar H.T.1998. Memon N, "Image Steganography: Concepts and Practice", WSPC/Lecture Notes Series, April 22, 2004. Anderson R.J, Petitcolas F.A.P., "On the Limits of Steganography", IEEE Journal of Selected Areas in Communications, 16(4):474-481, Special Issue on Copyright & Privacy Protection. ISSN 0733–8716

Lampson B.W.1973. "A Note on the Confinement Problem", Communications of the ACM, vol. 16. no. 10, pp. 613-615.

Memon N., Wong, P.1998."Protecting digital media content", Communications of the ACM, vol 41, no. 7 , pp. 34–43.

Microsoft Corporation.1990. "Microsoft Windows Programmer's Reference", vol. 2, v3, Microsoft Press, Redmond, WA

Murray A.H., Burchfiled R.W (eds.).1933. "The Oxford English Dictionary: Being a Corrected Re-issue", Oxford, England: Clarendon Pres.

Newman B.1940. "Secrets of German Espionage", London: Robert Hale Ltd.

Petitcolas F.A.P., Anderson R.J., Kuhn M.G.1999. "Information Hiding–A Survey", Proceedings of the IEEE, Special Issue on Protection of Multimedia Content, 87(7):1062-1078.

Pfitzmann B.1996. "Information Hiding Terminology", In Anderson [3], pp. 347-350, ISBN 3-540-61996-8.

Popa R.1998. "An Analysis of Steganographic Techiques", Ph.D Thesis.

Schott G.1655. "Schola steganographica: in classes octo distributa....", Jobus Hertz, printer, 1680, bound with: Gasparis Schotti...Technica curiosa... Herbipoli, [Courtesy of Whipple Science Museum, Cambridge.].

Simmons G.1984, "The Prisoners' Problem and the Subliminal Channel", CRYPTO83 Advances in Cryptology, pp. 51-67.

Singh P, Chaudhary R, Agarwal A.2012. "A Novel Approach of Text Steganography based on null spaces". IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661 Volume 3, Issue 4 (July-Aug. 2012), PP 11-17

Swanson M.1998. Kobayashi M., Tewfik A., "Multimedia Data Embedding and Watermarking Technologies", Proc. of IEEE, vol. 86, no. 6, pp. 1064-1087.

Tacticus A.1990. "How to Survive Under Siege / Aineias the Tactician", Oxford, England: Clarendon Pres, pp. 84-90 and 183-193, Clarendon Ancient History Series.

Wang H., Wang S.1984."Cyber Warfare: Steganography vs. Steganalysis", Communications of the ACM, vol. 47, no. 10, October 2004.

Zim H.S.1948. "Codes and Secret Writing", William Morrow, New York.

Atıcı, M. 2007. Steganografik yaklaşımların incelenmesi, tasarımı ve geliştirilmesi. *Yüksek Lisans Tezi,* Gazi Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Ankara

Caner, H. 2006. FPGA donanımı üzerinde araç plakası tanıma sistemi. *Yüksek Lisans Tezi,* Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Elektrik ve Elektronik Mühendisliği Anabilim Dalı, Ankara.

Gürel, H. 2006. Sayısal resim içerisine veri gizleme uygulamaları, *Yüksek Lisans Tezi,* Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Elektronik ve Bilgisayara Eğitimi Anabilim Dalı, Kocaeli.

Öcal, F. 2006. Güvenli iletişim için FPGA kullanarak şifreleme sistemi tasarımı ve gerçekleştirilmesi. *Yüksek Lisans Tezi,* Gazi Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar Eğitimi Anabilim Dalı, Ankara.

Şahin, A. 2007. Görüntü steganografide kullanılan yeni metodlar ve bu metodların güvenilirlikleri. *Doktora Tezi,* Trakya Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Edirne.

Ascii Tablosu,
http://www.antrak.org.tr/gazete/062006/cizimler/tolgatastan/ascii_table.jpg Erişim Tarihi (10.10.2012).

Xilinx,
http://www.xilinx.com/company/gettingstarted/index.html Erişim Tarihi (08.09.2012)
Fpga Hakkında Herşey,
http://www.fpganedir.com/ Erişim Tarihi (01.10.2012)

ÖZGEÇMİŞ

Adı Soyadı : Zeynep PEKER

Doğum Yeri ve Tarihi : BURSA – 08.09.2012

Yabancı Dili : İngilizce

Eğitim Durumu (Kurum ve Yıl)

Lise : Bursa Osmangazi Y.D.A Lisesi – 2005

Lisans : Uludağ Üniversitesi Elektronik Mühendisliği – 2010

Yüksek Lisans : UÜ Fen Bilimleri Enstitüsü Elektronik Mühendisliği A.B.D – 2013

Çalıştığı Kurum ve Yıl : BUZ İletişim Ltd.Şti. Hizmetleri – (2011-…)

İletişim : zeynep_pkr@hotmail.com