İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

DES BLOK ŞİFRELEME ALGORİTMASININ FPGA ÜZERİNDE DÜŞÜK ENERJİLİ TASARIMI

YÜKSEK LİSANS TEZİ

Tarık KAPLAN

Anabilim Dalı : Elektronik ve Haberleşme Mühendisliği

Programı : Elektronik Mühendisliği

Tez Danışmanı: Yrd. Doç. Dr. Sıddıka Berna ÖRS YALÇIN

Mayıs 2009

ÖNSÖZ

Yüksek lisans tezimin ortaya çıkmasında önemli katkıları bulunan, bana her zaman yol gösteren ve beni teşvik eden danışman hocam Yrd. Doç. Dr. S. Berna ÖRS YALÇIN' a çok teşekkür ederim.

Ayrıca yardımlarını benden esirgemeyen arkadaşım Ali Can ATICI' ya ve beni hayatım boyunca her yönden destekleyen ve hiçbir zaman yalnız bırakmayan aileme şükranlarımı sunarım.

Son olarak, tez çalışmalarım sırasında maddi destekte bulunan TÜBİTAK'a teşekkür ederim.

Mayıs 2009

Tarık KAPLAN

Elektronik Mühendisi

İÇİNDEKİLER

Sayfa

KISALTMALAR

AES	**:** Advanced Encryption Standard
DES	**:** Data Encryption Standard
FPGA	**:** Field Programmable Gate Array
LUT	**:** Look-Up Table
RAM	**:** Random Access Memory
ROM	**:** Read-Only Memory
FIPS	**:** Federal Information Processing Standards
NIST	**:** National Institute of Standards and Technology
ASIC	**:** Application Specific Integretd Circuit
HDL	**:** Hardware Description Language
CLB	**:** Configurable Logic Blocks
SRAM	**:** Static Random Access Memory
EPROM	**:** Erasable Programmable Read Only Memory
EEPROM	**:** Electrically Erasable Programmable Read Only Memory
PLD	**:** Programmable Logic Devices

ÇİZELGE LİSTESİ

Sayfa

ÇİZELGE LİSTESİ
hazırlanırken 1 satır boşluk bırakılır.

Bir satırı aşan isimlerde satırların burada olduğu gibi aynı hizadan başlamasına özen gösteriniz.

Bu bir nottur, çıktı almadan önce siliniz.

ŞEKİL LİSTESİ

Sayfa

DES BLOK ŞİFRELEME ALGORİTMASININ FPGA ÜZERİNDE DÜŞÜK ENERJİLİ TASARIMI

ÖZET

Son yıllarda… Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

ÖZET hazırlanırken 1 satır boşluk bırakılır.

Bu genişletilmiş özet 200 kelime ile 5 sayfa arasında olur.

Türkçe tezlerde Türkçe özetin önce olması önerilir.

Bu bir nottur, çıktı almadan önce siliniz.

ENERGY EFFICIENT FPGA IMPLEMENTATION OF DES ALGORITHM

SUMMARY

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

SUMMARY hazırlanırken 1 satır boşluk bırakılır.

This is an extended summary which is min 200 words to max 5 pages

Bu bir nottur, çıktı almadan önce siliniz.

Giriş bölümü tek numaralı sayfadan başlaması gerekmektedir. Bir sonraki sayfa çift numaralı olacağı için 1 tane boş sayfa ekledik.

Bu bir nottur, çıktı almadan önce siliniz.

1. GİRİŞ

Günümüz teknoloji dünyasında veri aktarımı önemli konulardan biridir. Özellikle veri aktarımının güvenli bir şekilde yapılması, üzerinde en çok durulan konuların başında gelir. Gün geçtikçe çok büyük miktarlardaki verilerin bilgisayarlar tarafından işlenmesi, saklanması ve elektronik haberleşme kanalları üzerinden bir yerden diğer bir yere iletilmesi gündelik hayatın sıradan işlerinden biri haline gelmektedir. Ancak verilerin iletimi sırasında kullanılan haberleşme kanallarının herkesin kullanımına ya da erişimine açık olması, sözkonusu verilerin yetkili olmayan (üçüncü) şahıslar tarafından değiştirilmesi, yok edilmesi ve içeriğine ulaşılması problemini gündeme getirmektedir. Bu noktada, mesajların herkesin erişimine açık elektronik haberleşme kanallarından iletilebilmesi için bir takım dönüşümler sonucunda değişikliğe uğratılarak üçüncü şahıslar için anlaşılamaz bir hale getirilmesi gerekmektedir. Bu amaçla yapılan tüm işlemlere birden Kriptografi ya da Şifreleme adı verilir [1]. Diğer bir tanımla Kriptografi, en az iki kişinin güvenli olmayan bir kanal üzerinden üçüncü bir şahsa bilgi sızdırmadan haberleşmesini sağlamak amacıyla matematiksel tekniklerin geliştirilmesidir [2].

Kriptografi, bilgiyi yani veriyi güvenli bir şekilde sadece istenilen kişiye ulaştırmak amacıyla uzun süredir kullanılmaktadır. İlk başlarda askeri amaçlı kullanıldıysa da gelişen teknoloji ile birlikte ortaya çıkan güvenlik açığını kapatmak ve bilginin güvenilir bir şekilde taşınmasını sağlamak amacıyla sivil yaşamda da yerini almıştır. Veri transferi sırasında güvenilirliği sağlamak için sıkça kriptografik algoritmalar kullanılır. Kullanılan bu algoritmaları donanımsal veya yazılımsal olarak gerçeklemek mümkündür. Yazılımsal gerçeklemeler daha az maliyet getirmekle birlikte yavaş ve güvensizdirler. Donanımsal gerçeklemelerin ise istenilen yüksek hızlarda çalışma imkanları vardır, ayrıca yazılımsal gerçeklemelere oranla daha güvenilir gerçeklemelerdir [3].

Pek çok kripto algoritması yüksek hız ve yüksek işlem hacmine sahip olacak şekilde Sahada Programlanabilir Kapı Dizileri (FPGA) ve Uygulamaya Özel Tümdevre (ASIC) teknolojileri kullanılarak gerçeklenmektedirler. Tasarımlarda belirlenen hedeflere göre algoritmaların gerçeklenmesi farklı şekillerde olabilmektedir. Kimi tasarımlar hızı ön plana koyarken, kimi tasarımlar alandan tasarrufu hedeflemektedir. Bunların dışında, güvenliği, düşük güç tüketimini veya düşük enerji harcamayı hedefleyen devre tasarımları olabilmektedir. Doğal olarak her tasarımcının hedefi, gerçeklediği algoritmayı en iyi şekilde yapmaktır. Ancak bahsedilen hedefleri çoğu zaman aynı anda yakalamak mümkün değildir. Bu sebeple Kripto algoritmalarının tasarımı belirlenen hedefe yönelik çalışılarak gerçeklenmektedir.

Düşük enerji hedeflenen bir Kripto algoritması tasarımında, dikkat edilmesi gereken konular güç ve zamandır. Devrelerin harcadığı enerji, tükettiği güç ve şifreleme için kullanılan süreyle doğru orantılıdır. Bir elektronik devrede temel olarak harcanan iki tür güç kavramından bahsedilebilir. Bunlardan ilki statik güç olarak adlandırılır, devreye herhangi bir giriş uygulanmadan, devrenin sükunet halinde harcadığı güçtür. Devre üzerinde harcanan diğer güç ise dinamik güçtür. Dinamik güç devreye verilen girişlerin işlenmesi sırasında harcanan güçtür. Bir devreye ait statik güç harcaması zamanla değişmezken, dinamik güç harcaması uygulanan girişlere bağlı olduğundan zamanla değişmektedir. Enerjiyi etkileyen diğer faktör olan şifreleme zamanı da devrenin çalışma frekansına ve algoritmanın gerçeklenme şekline bağlı olarak değişim göstermektedir. Düşük enerji tüketen devre elde edebilmek için hem düşük güç tüketimi hem de yüksek işlem hacmi gerekmektedir.

1.1 Tezin Kapsamı

Veri Kodlama Standardı (Data Encryption Standart (DES)), 1977 yılında Amerikan Ulusal Standartlar ve Teknoloji Enstitüsü (National Institute of Standards and Technology (NIST)) tarafından standart olarak kabul edilmiş ve Federal Bilgi İşleme Standardı (Federal Information Processing Standards (FIPS)) olarak yayınlanmıştır [4]. DES algoritması uzun yıllar standart olarak kullanılmış, ancak 1999 yılında yerini Gelişmiş Kodlama Standardı (Advanced Encryption Standard (AES))' e bırakmıştır. DES algoritması, bilgiyi şifrelemek ve çözmek için kullanılan simetrik bir blok şifreleyicidir.

DES blok şifreleme algoritması, en çok bilinen blok şifreleme algoritmalarından biridir. Günümüzde çok yüksek güvenlik gerektirmeyen alanlarda halen kullanılmaktadır. Kablolu modemler, uzaktan erişim sunucuları, şifreli veri depolama aygıtları örnek kullanım alanları olarak verilebilir [5]. Literatürde pek çok yazılımsal ve donanımsal DES gerçeklemesi mevcuttur. Gelişen teknoloji ile birlikte kaynak kısıtlı ortamlarda da bilgiyi şifreleme gereksinimi duyulmaktadır. Bu ortamlarda enerji tasarruflu DES gerçeklemelerine ihtiyaç duyulmaktadır.

Bu çalışmada, DES blok şifreleme algoritmasının düşük enerjili tasarımı hedeflenmiştir. Güç tasarrufu yöntemlerinden ve işlem hacmini arttırma yöntemlerinden yararlanılarak, farklı DES yapıları gerçeklenmiştir. Literatürdeki diğer DES gerçeklemeleri de incelenerek, en düşük enerji tüketen DES devresi gerçeklenmiştir. Tüm bu tasarımlar sırasında Sahada Programlanabilir Kapı Dizileri (FPGA - Field Programmable Gate Array) teknolojisi kullanılmıştır.

1.2 Tezin Konuya Katkısı

Literatürde, DES gerçeklemelerinde genellikle yüksek hız hedef olarak alınmıştır. Düşük güç veya düşük enerji hedefiyle tasarım yok denecek kadar azdır. Bu çalışma ile hem literatürdeki DES gerçeklemeleri alan, zaman, iş hacmi, güç ve enerji açısından karşılaştırılmış hem de enerji tüketimi en düşük devre gerçeklemesi sunulmuştur. Ayrıca düşük güç tüketen devre tasarımı için gerekli yapısal değişiklikler ve yüksek hızlı devre için gerekli yapısal değişiklikler üzerinde durulmuştur. Bununla birlikte, DES algoritmasını gerçeklemek isteyen tasarımcılar için kapsamlı bir kaynak hazırlanmaya çalışılmıştır.

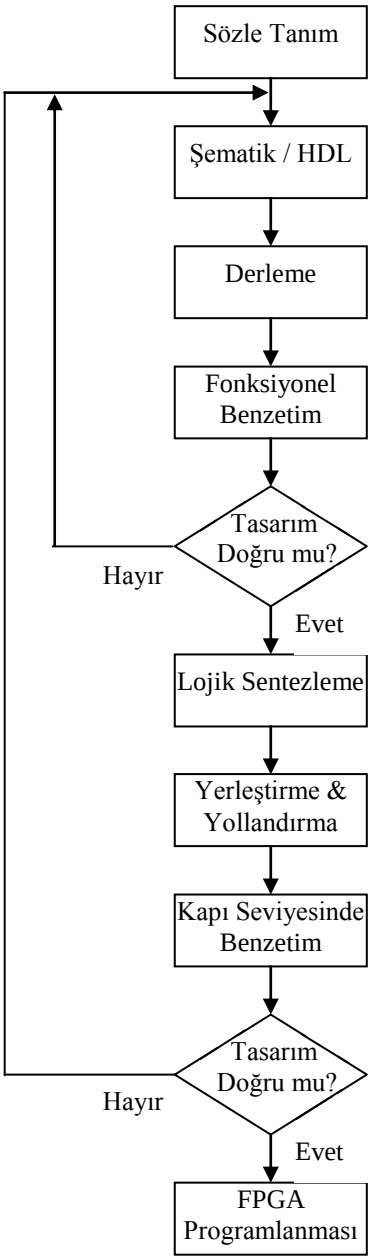
2. SAHADA PROGRAMLANABİLİR KAPI DİZİLERİ

Sahada Programlanabilir Kapı Dizileri (Field Programmable Gate Array, FPGA) yaygın olarak kullanılan programlanabilir devre elemanlarıdır. Programlanabilir devre elemanları, geniş uygulama alanları sağlayabilmek için genel amaçlı tümdevreler olarak tasarlanmışlardır. Programlanabilir eleman ve arabirimler VE (AND), VEYA (OR), ayrıcalıklı VEYA (E-XOR), DEĞİL (NOT) işlemlerini veya daha karmaşık olan dekoder, çoklayıcı gibi matematiksel işlemleri gerçekleştirmek amacıyla programlanabilir. FPGA yapısında bulunan arabirimler tasarımdaki bağlantılar göz önüne alınarak elektriksel olarak programlanabilir ve istenildiği kadar programlanabilme yapısına sahip olduğundan tasarımlarda büyük kolaylık sağlar. Pek çok FPGA yapısı programlanabilir eleman ve ara birimlere ek olarak hafıza birimleri de bulundurur. Bu hafıza birimleri ayrık flip-flop yapılarından veya hafıza bloklarından oluşabilir [3].

Programlanabilir yapılar, örneğin FPGA'lar ASIC tasarımlara oranla daha yavas çalışmalarına rağmen tekrar programlanabilme özelliği ve tasarımların basit kontrol edilebilir olması nedeniyle tasarımın daha ucuza mal edilebilmesi açısından önemli avantajlara sahiptir.

FPGA' ların programlanması aşamasında ilk olarak tasarlanacak devrenin sözle tanımı verilir. Daha sonra şematik olarak veya yüksek seviyeli donanım tanımlama dilleri (Hardware Description Language, HDL) kullanılarak tasarım yapılır. Tasarım her ne şekilde olursa olsun derleme işleminden sonra devreye ait standart bağlantı listesi (netlist) oluşturulur. Yapılan tasarımın devreye ait istenen özellikleri yerine getirip getirmediği fonksiyonel benzetim (functional simulation) yapılarak test edilir. Benzetim sonucuna göre gerekirse, tasarımda değişiklikler yapılarak istenen sonuç elde edilene kadar bu şekilde iterasyona devam edilir. İstenilen sonuç elde edildikten sonra lojik sentezleme adımına geçilir. Bu aşamada üretilecek devre gerçeklenirken kullanılacak FPGA seçilir. Buna göre lojik sentezleyicinin, kullanılacak FPGA' yı desteklemesi gerekmektedir. Sentezleme işleminde ayrıca varsa, devreye ait lojik kısıtlamalar (Giriş / Çıkış bacakları, zamanlama, yerleştirme, saat frekansı, kritik

yollar gibi) kullanıcı kısıtlama dosyası (user constraints file, Xilinx) ile birlikte verilebilir. Lojik sentezleyiciler, istenilen fonksiyonların en iyi şekilde gerçeklenebilmesi için gerekli lojik indirgemeleri (logic optimization) de yaptıktan sonra elde edilen lojik fonksiyonların FPGA içerisindeki lojik bloklarla eşleştirilmesi işlemi yapılarak (technology mapping) kapı seviyesinde bir bağlantı listesi oluşturulur. Teknoloji eşleştirilmesi sırasında, kullanıcı kısıtlama dosyası da kullanılarak zamanlama gereksinimi karşılanmak amacıyla gerekirse daha fazla lojik eleman kullanılabilir. Sentezleme sonrasında, yerleştirme ve yollandırma (placement and routing) işlemleri yapılır. Bu adımda, devre fonksiyonları ile eşleştirilmiş lojik bloklar FPGA içerisinde uygun yerlere yerleştirilir ve bu bloklar arasındaki bağlantılar oluşturulur. Lojik yolların daha kısa olması amacıyla birbirleriyle ilişkili CLB (Configurable Logic Blocks)' ler yakın yerleştirilir. Yollandırmada ise bağlantılar uygun şekilde seçilir. Örneğin, tasarımın bir çok alanında bir işarete ihtiyaç varsa en küçük gecikmeyi sağlamak amacıyla uzun bir yol kullanılır. Bu aşamadan sonra kapı seviyesinde benzetimin gerçekleştirilmesi uygun olacaktır. Çünkü artık bütün CLB' lere (LUT veya Çoğullayıcılar ve flip-flop' lara) ve yollandırma bağlantılarına ait gecikmeler gerçeğe çok yakın olarak elde edilmiştir. Bu gecikmeler de eklenerek devrenin benzetimi yapıldığında, zamanlama ve hız açısından kritik yollar saptanabilir. Yerleştirme ve yollandırma sonrası benzetimlerde istenilen sonuçlar elde edildikten sonra FPGA' nın programlanması aşamasında kullanılacak bit dizisi, üreticinin sağladığı yazılımla elde edilir. FPGA' nın uygun donanım kullanılarak programlanmasıyla tasarım tamamlanır. Şekil 2.1'de yüksek seviyeli tasarım sürecine ilişkin akış diyagramı verilmiştir [2].



Şekil 2.1 : Tasarım sürecine ilişkin akış diyagramı.

2.1 FPGA' ların Güç Tüketim Özellikleri

FPGA' ların güç tüketim özelliklerinden bahsetmeden önce güç kavramını anlamak gerekir. Devrelerde harcanan toplam güç, statik ve dinamik güçlerin toplamıdır. Statik güç, devrenin sükunet halinde harcadığı güçtür. Dinamik güç ise devredeki elemenların çıkışlarının 0'dan 1'e veya 1'den 0'a lojik geçişleri sırasında harcadıkları güçtür. Statik güç, devre çıkışındaki lojik geçişler ve devrenin çalışma frekansından bağımsızdır. Halbuki dinamik güç, kullanılan devre yapısına gore farklılık gösterir. Bu yüzden çalışma boyunca kullanılan güç kavramı, aksi belirtilmediği sürece, dinamik gücü temsil etmektedir.

Günümüzde en çok kullanılan programlama teknolojileri SRAM (Static Random Access Memory), anti-sigorta ve FLASH programlamadır. Bunların dışında, sigorta, EPROM (Erasable Programmable Read Only Memory) ve EEPROM (Electrically Erasable Programmable Read Only Memory) teknolojileri de mevcuttur. Ticari amaçlı kullanılan bazı FPGA'lar Çizelge 2.1'de verilmiştir.

Çizelge 2.1 : Bazı ticari FPGA'lar.

Üretici	**Mimari**	**Lojik Blok Tipi**	**Programlama Teknolojisi**
Actel	Satır Bazlı	Çoğullayıcı Bazlı	Anti-Sigorta
Altera	Hiyerarşik PLD	PLD Blok	EPROM
QuickLogic	Simetrik Dizi	Çoğullayıcı Bazlı	Anti-Sigorta
Xilinx	Simetrik Dizi	Doğruluk Tablosu	Statik RAM

Anti-sigorta tabanlı FPGA'lar sadece bir kez programlanabilir. Bu cihazlar, programlandıktan sonra birer ASIC gibi ele alınabilir. Ancak enerjileri kesildiğinde, SRAM tabanlı FPGA'ların aksine kurgularını korurlar. SRAM tabanlı FPGA'lar ise tekrar tekrar programlanabilmektedir. Bu cihazlar, her ilklendirmede, kurgulama belleğine, kurgu bilgisini yeniden yükler. Enerjileri kesildiğinde ise kurgulama belleği silinir. Bu sayede, sistemin donanımsal tasarımını değiştirmeden, sadece kurgulama dosyası değiştirilerek, sistemin işlevi değiştirilebilir. Bu özellik, araştırma-geliştirme uygulamalarında büyük kolaylık sağlamakta ve sistem maliyeti düşürmektedir. Ürünün çalıştığı ortamda güncellenebilmesi SRAM FPGA'ların seçilmesindeki en önemli etkendir [6-8].

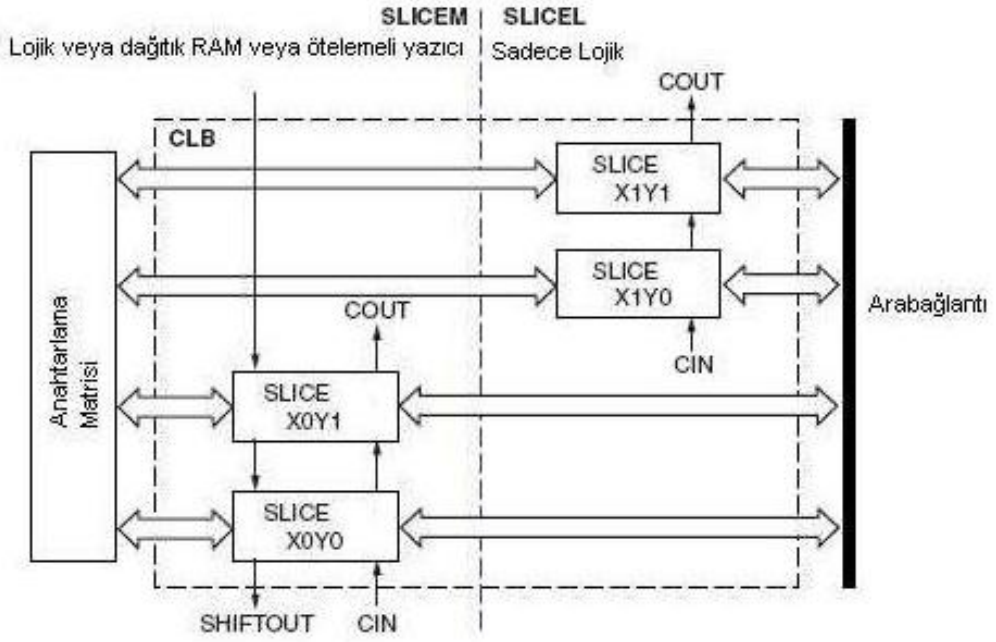
Kullanılan programlama teknolojisine göre devrelerin harcadığı güç değişmektedir. SRAM teknolojisinde her ilklendirmede tekrar kurgu yükleme bilgisine bağlı olarak güç harcaması anti-sigorta teknolojisine oranla daha fazladır. Ayrıca anti-sigorta FPGA'lar ara bağlantı noktalarındaki düşük çıkış kapasiteleri nedeniyle SRAM FPGA'lara oranla çok daha az miktarlarda dinamik güç harcar. SRAM tabanlı FPGA'larda dışardan bağlanan EPROM veya Flash bellek ile konfigürasyon sağlanır. Ayrıca konfigürasyon kurulumu için de güç tüketimi yapılmaktadır. Bu bilgiler ışığında SRAM tabanlı FPGA'ların anti-sigorta tabanlı FPGA'lara oranla çok daha fazla güç tükettikleri açıktır [3]. Düşük güç tüketim avantajına sahip olan anti-sigorta tabanlı FPGA'lar belirlenmiş bir tasarıma uygun olarak üretildikleri için başka bir tasarım için kullanılmaları mümkün değildir. Bu yüzden yapılan çalışma sırasında anti-sigorta tabanlı FPGA'lar düşük güç tüketecekleri bilindiği halde seçilmemişlerdir. Seçilen FPGA SRAM tabanlıdır. Çalışma süresince, SRAM tabanlı FPGA'ların güç tüketim dezavantajları göz önüne alınmıştır.

2.2 Gerçekleme Sırasında Kullanılan FPGA

DES Blok Şifreleme Algoritmasının FPGA üzerinde düşük enerjili tasarımı çalışmasında Xilinx firmasına ait Spartan-3 XC3S5000 FPGA'sı kullanılmıştır. XC3S5000 FPGA'sı çok büyük bir FPGA'dır. 633 adet giriş çıkış ve 8320 adet konfigüre edilebilir lojik bloğa kadar destek sağlayabilmektedir [9]. Çalışma sırasında farklı DES yapıları gerçeklenmiş ve karşılaştırılmıştır. Çalışma başında Xilinx firmasına ait VirtexE XCV1000E FPGA'sı kullanılmaya çalışılmış fakat tasarlanan devrede 194 adet giriş çıkış olduğundan ve XCV1000E FPGA'da en fazla

162 adet giriş çıkış desteklendiğinden bu FPGA kullanılamamıştır. Daha sonra VirtexE XCV2000E FPGA'sı seçilerek tasarım yapılabilmiştir. XCV2000E FPGA'da 404 adet giriş çıkış desteklendiğinden tasarıma uymuştur. Ancak XCV2000E FPGA kullanılarak tasarlanın devrenin XPOWER güç ölçüm aracı ile yapılan güç analizinde besleme gerilimi sükunet değeri sıfır çıkmıştır. Yapılan güç analizlerini etkilememesi amacıyla sükunet besleme geriliminin gerçek değerini verebilen Spartan-3 serisinin en gelişmiş versiyonu olan XC3S5000 FPGA'sı seçilerek bütün DES yapıları gerçeklenmiş ve karşılaştırmalar adil bir şekilde yapılmıştır. Gerçekleme aşamasında VHDL dili kullanılmış olup, FPGA' da sentezleme, yerleştirme ve yollandırma aşamaları için Xilinx ISE 9.2i programı kullanılmıştır. Ayrıca simülasyon aşamasında Modelsim 6.3c ve güç ölçümleri için de XPOWER aracı kullanılmıştır.

Diğer FPGA'larda olduğu gibi XC3S5000'de konfigüre edilebilir lojik blok yapısı (CLB), giriş çıkış bağlantıları ve Blok RAM'lardan oluşmaktadır. CLB yapıları RAM tabanlı LUT (Look-Up Table)' ler içerir, kaydedici olarak veya lojik işlemler için kullanılır. Bu bloklar genel yollandırma hattına bağlanmaktadır. Her bir CLB için 4 ara bağlantı dilimi bulunur, bu ara bağlantı dilimleri SLICEM ve SLICEL olarak adlandırılır. Sekil 2.2'de bu yapı verilmektedir. Giriş/Çıkış blokları, giriş/çıkış pinleri ile iç yapıdaki elemanlar arasındaki veri akışını kontrol eder. Her bir giriş/çıkış bloğu çift yönlü iletimin yanında 3-durumlu çıkış kontrolünü de destekler. Blok RAM'lar ise veri depolama görevini üstlenirler. XC3S5000 FPGA'sı 4 adet Blok RAM sütununa sahiptir [9].



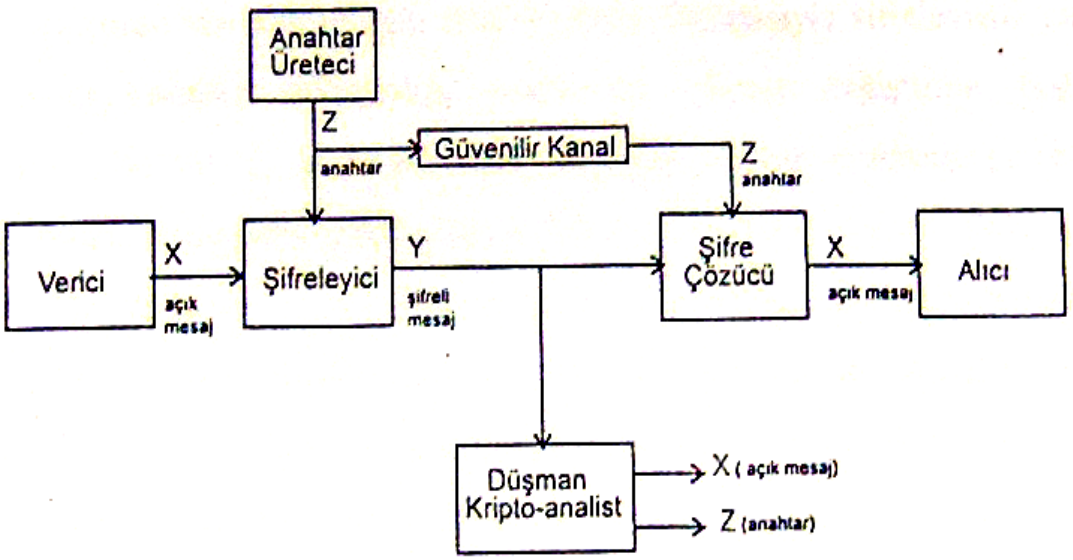
Şekil 2.2 : Spartan 3 CLB Ara Bağlantıları.

3. DES BLOK ŞİFRELEME ALGORİTMASI ve GERÇEKLENMESİ

3.1 Blok Şifreleme Sistemleri

Günümüzde şifreleme sistemleri, şifreleme anahtarının gizli tutulduğu "gizli-anahtarlı" ve şifreleme anahtarının açık olduğu "açık-anahtarlı" sistemler olmak üzere ikiye ayrılırlar. Bu sistemler sırasıyla "simetrik" ve "anti-simetrik" şifreleme sistemleri olarak da bilinirler. Açık-anahtarlı ya da anti-simetrik şifreleme sistemlerine örnek RSA algoritmasının kullanıldığı sistemlerdir. Bu tür sistemlerde şifreleme ve şifre çözme işlemleri, gizli-anahtarlı sistemlerin aksine, farklı anahtarlar kullanılarak gerçekleştirilir. Bu nedenle açık-anahtarlı sistemler simetrik olmayan bir yapıya sahiptir [10].

Simetrik yapılı gizli-anahtarlı şifreleme sistemlerinde şifreleme ve şifre çözme anahtarları aynıdır. Şekil 3.1 'de gizli-anahtarlı sistemin genel yapısı görülmektedir:



Şekil 3.1 : Gizli-anahtarlı şifreleme sisteminin genel görünüşü.

Sistemde *E* şifreleme, *D* de şifre çözme dönüşümlerini göstermektedirler. Şifreleme ve şifre çözme işlemleri aynı anahtarı kullanılarak gerçekleştirilir:

$Y = E_Z(X)$ (Şifreleme) **(3.1)**

$X = D_Z(Y)$ (Şifre çözme) **(3.2)**

Düşman kripto-analistin gizli anahtara erişimi güvenilir bir kanalla engellendiği taktirde sistem oldukça güvenilir olabilmektedir.

Gizli anahtar kullanan simetrik şifreleme sistemleri iki gruba ayrılır: Dizi Şifreleme (Stream Cipher) ve Blok Şifreleme (Block Cipher) sistemleri.

"Dizi şifreleme" sistemleri her bir mesaj birimini, zamanla değişen bir fonksiyon kullanarak şifreler. Dizi şifreleyiciler yüksek hızlı iletişim için en iyi alternatiflerden biridir. Yüksek hatalı iletişim ortamlarında hata riskini azalttığından tercih sebebidirler [3]. Dizi şifreleme sisteminin iç yapısı zamana bağlı olarak durum değiştiren bir makine gibi düşünülebilir. Dolayısıyla şifrelemede kullanılan fonksiyonun zamana bağımlılığı makinenin durum değiştirme bağıntısıyla tanımlanır. Bu tür bir sistemde mesaj biriminin büyük olmasına gerek yoktur. Aksine mesaj biriminin olabildiğince küçük tutulması istenir. Dizi şifreleme sistemlerinde kullanılan mesaj birimleri genelde Latin alfabesinden bir karakter ya da tek dijitlik sayılardır. Bir dizi şifreleme sisteminde mesaj dijitleri tek tek şifrelendiğinden mesajlar genelde bir dizi şeklinde düşünülür. Mesaj dizisinin her bir dijiti şifrelendikten sonra sistem belirli bir kural uyarınca durum değiştirir. Sistemin durum değiştirmesi şifreleme anahtarının da değişmesine neden olur. Bu nedenle dizi şifreleme sistemlerinde anahtar da mesajla aynı boyutta bir dizi şeklindedir. Bu anahtar dizisinin yapısı şifreleme sisteminin güvenilirliği konusunda oldukça büyük bir önem taşımaktadır.

İkinci simetrik şifreleme sistemi olan "Blok Şifreleme" sistemi, en temel anlamıyla basit yerine koyma sistemleridir. Blok Şifreleyiciler, sabit uzunluktaki bloklar halinde aldıkları açık veriyi, yine aynı uzunluktaki bloklar halinde şifrelenmiş veriye çeviren simetrik anahtarlı şifreleme algoritmalarıdır. Bu dönüştürme işlemi kullanıcı tarafından belirlenen gizli bir anahtar kullanılarak yapılır. Şifre çözme işlemi de, yine sabit uzunluktaki bloklar halinde alınan kapalı verinin bu kez ters dönüşümden geçirilerek, bloklar halinde açık veriye dönüştürülmesiyle gerçekleştirilir [8].

Blok şifreleyiciler günümüzde şifreleme işlemlerinde yaygın olarak kullanılmaktadır. En çok bilinenlere örnek olarak; 64 bitlik blokları kullanan DES ve 128 bitlik

blokları kullanan AES algoritmaları gösterilebilir. Şifrelenecek olan verinin uzunluğu blok uzunluğundan daha büyük olması durumunda veri blok uzunluklarına parçalanır ve şifreleme işlemi gerçekleştirilir. Veriyi blok uzunluklarına parçaladıktan sonra şifreleme işlemi Elektronik Kod Kitabı (ECB), Şifre Bloklarını Zincirleme (CBC), Şifreyi Geri Besleme (CFB) ve Çıkışı Geri Besleme (OFB) gibi yöntemlerle gerçeklenebilir. Seçilen yöntem şifrelenmiş veri üzerinde çok büyük bir etkiye sahiptir. Aynı veri bloğunun, aynı anahtar bloğu ile şifrelenmesi sonucu hep aynı çıkış bloğu oluşur, bu nedenle şifreleme işleminde güvensizlik oluşabilir. Kullanılan farklı yöntemler sayesinde şifreleme işlemi daha güvenilir hale getirilir [3].

Bu çalışmada, DES Blok Şifreleme algoritması en çok bilinen yöntem olan Elektronik Kod Kitabı (ECB) yöntemiyle gerçeklenmiştir. ECB yönteminde giriş verisi olarak gelen her düz metin bloğu diğer bloklardan bağımsız olarak şifrelenir veya çözülür.

3.2 DES Blok Şifreleme Algoritması

Günümüzde teknolojinin sürekli geliştiği ve çok hızlı bir şekilde gelişmeye devam edeceği bilinen bir gerçektir. Gelişen teknoloji ile birlikte veri iletimi ve iletilen verinin güvenliği önemli bir unsur haline gelmiştir. Şifreleme algoritmaları, iletilen verinin güvenliği için kullanılan yöntemlerdir. Bu algoritmaların teknolojinin gerektirdiği şekilde olacağı, teknolojiye ayak uyduramayanların kullanımının terk edileceği, yeni algoritmaların bulunacağı bilinen bir gerçektir. Teknoloji ile gelişen bu algoritmalar standartlaşma ihtiyacı hissetmektedir. Bu yüzden birçok ülke kendi standartlaşma enstitüsünü kurmuştur. ABD Ulusal Standartlar ve Teknolojiler Ensititüsü (NIST) de bunlardan birisidir [11]. Bu bölümün devamında NIST' ın standart olarak kabul ettiği DES algoritması ayrıntılı olarak incelenecektir.

Veri Kodlama Standardı (Data Encryption Standart (DES)), 1977 yılında Amerikan Ulusal Standartlar ve Teknoloji Enstitüsü (National Institute of Standards and Technology (NIST)) tarafından standart olarak kabul edilmiş ve Federal Bilgi İşleme Standardı (Federal Information Processing Standards (FIPS)) olarak yayınlanmıştır [4]. İlk başlarda 10-15 yıl standart olarak kullanılacağı düşünülen DES algoritması, uzun yıllar standart olarak kullanılmış, ancak 1999 yılında yerini Gelişmiş Kodlama Standardı (Advanced Encryption Standard (AES))' e bırakmıştır [12]. DES algoritması, bilgiyi şifrelemek ve çözmek için kullanılan simetrik bir blok şifreleyicidir.

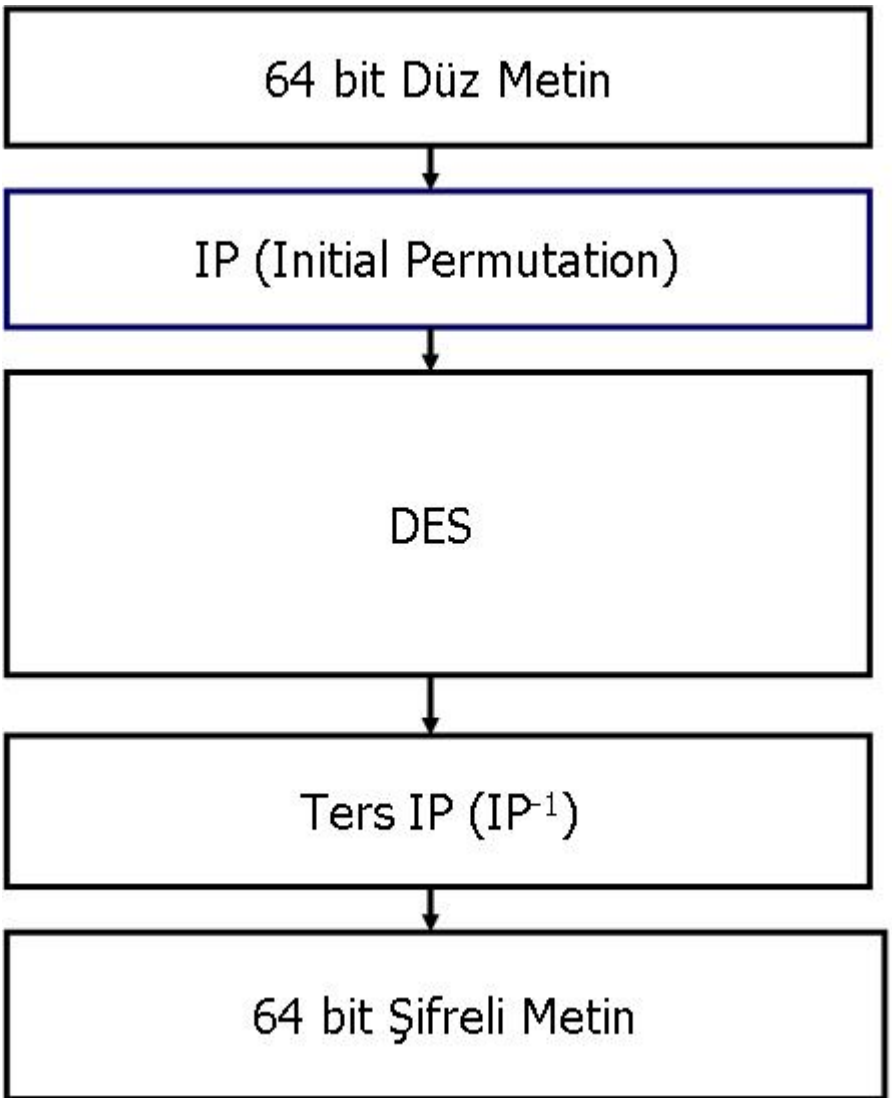
DES Blok Şifreleme Algoritması, 64 bitlik anahtar ile 64 bitlik düz metni 64 bitlik şifreli metne dönüştüren senkron bir blok şifreleme sistemidir. DES algoritması, daha önceden bulunmuş olan Feistel şifreleme yönteminin özel bir versiyonudur. Feistel tipi şifrelemede, her adımda veri iki eşit uzunluktaki blok halinde şifrelenir. Bu iki eşit blok genelde sol yarı ve sağ yarı blok olarak tanımlanır. DES algoritmasında, şekil 3.2'de görüldüğü gibi, şifrelenecek metin öncelikle IP (Initial Permutation – ilk permütasyon)' dan geçer. Daha sonra DES kutusundan geçer ve ters IP sonucunda şifreli metin elde edilir.

DES algoritması için kullanılan 64 bitlik anahtarın 56 biti algoritma içinde işlemden geçer. Kalan 8 bit benzerlik veya hata bulmak için kullanılabilir.

64 bitlik düz metin öncelikle IP' den geçer. Permütasyon işleminde, bitlerin yerleri değiştirilerek şifrelemenin ilk adımı gerçekleştirilir. İlk Permütasyondaki bit değişim sırası şöyledir [4]:

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7



Şekil 3.2 : DES Algoritması Genel Yapısı.

IP işleminden sonra girişin 58. biti ilk bit, 50. biti ise 2. bit olur ve bu şekilde devam eder. Permütasyondan geçen giriş bloğu daha sonra karmaşık anahtar bağımlı DES kutusuna girer. DES kutusu çıkışı ise ters ilk permütasyona (IP^{-1}) girer. Ters ilk permütasyon sırasındaki bit değişim sırası şöyledir [4]:

IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Burda da girişin 40. biti ilk bit, 8. biti ikinci bit olur ve böyle devam eder. Ters IP çıkışında ise şifreli metin elde edilmiş olur.

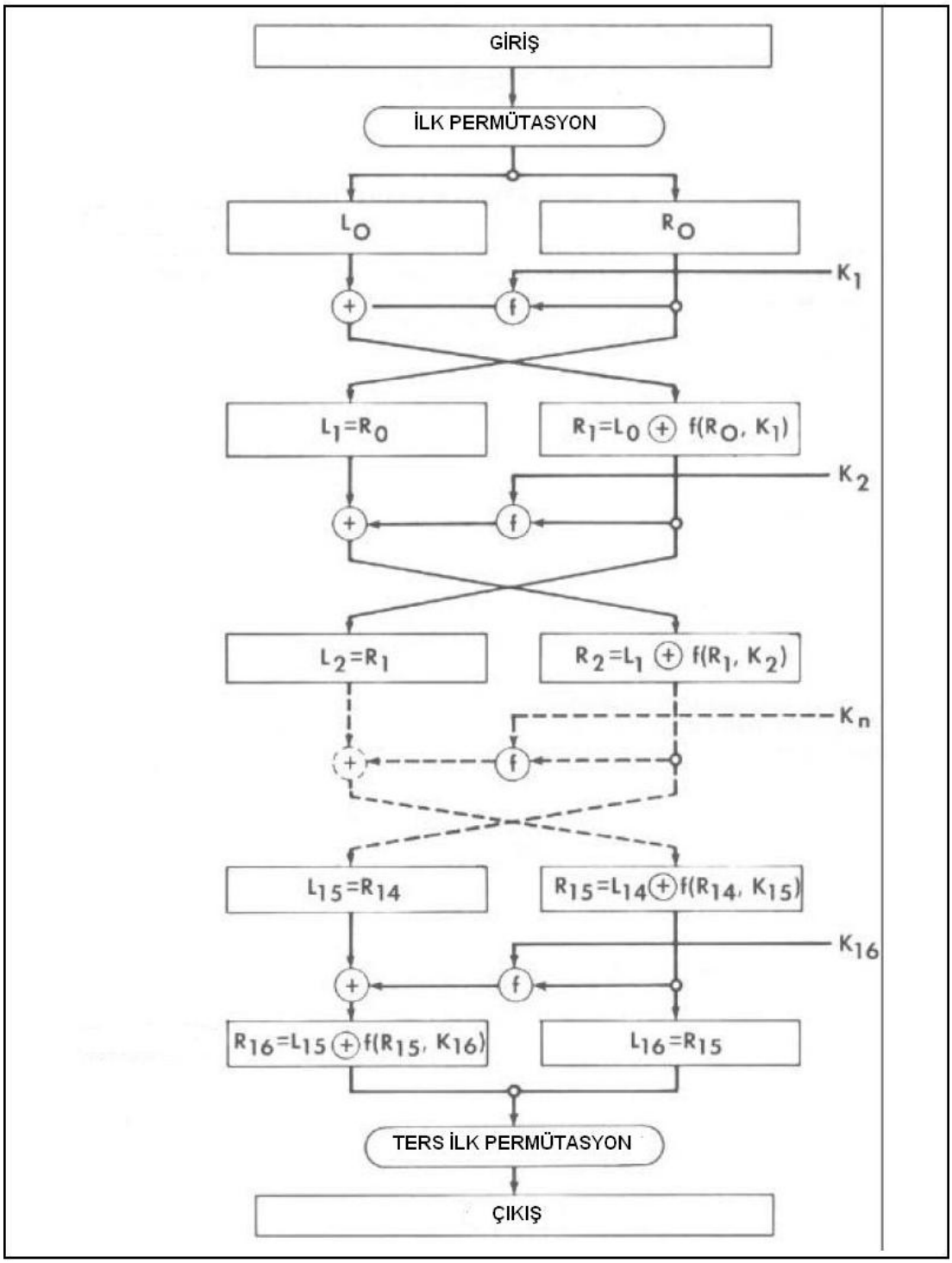
DES algoritması Feistel turu da denilen 16 turdan oluşur. Şekil 3.3' te DES kutusunun içeriği ve turların dağılımı görülmektedir.

İlk permütasyon çıkışında oluşan sağ ve sol yarı bloklar birinci tur işlemlerine girer. Sonraki turun sol yarı bloğu, var olan turun sağ yarı bloğudur. Sağ yarı blok ise anahtar ile birlikte DES *f* fonksiyonuna girer. DES *f* fonksiyonu çıkışı, sol yarım küre ile e-xor' lanarak sonraki turun sağ yarı bloğunu oluşturur. Denklem 3.3 ve denklem 3.4' te tur içindeki işlemler sembolize edilmiştir. L' ve R' sonraki turu ifade eder. Turlar bu şekilde devam eder ve 16. tur sonunda oluşan sol ve sağ yarı bloklar ters IP fonksiyonuna girer. Diğer turlardan farklı olarak 16. turdaki çıkışlar ters IP fonksiyonuna şekil 3.3'te görüldüğü gibi yer değiştirerek girer.

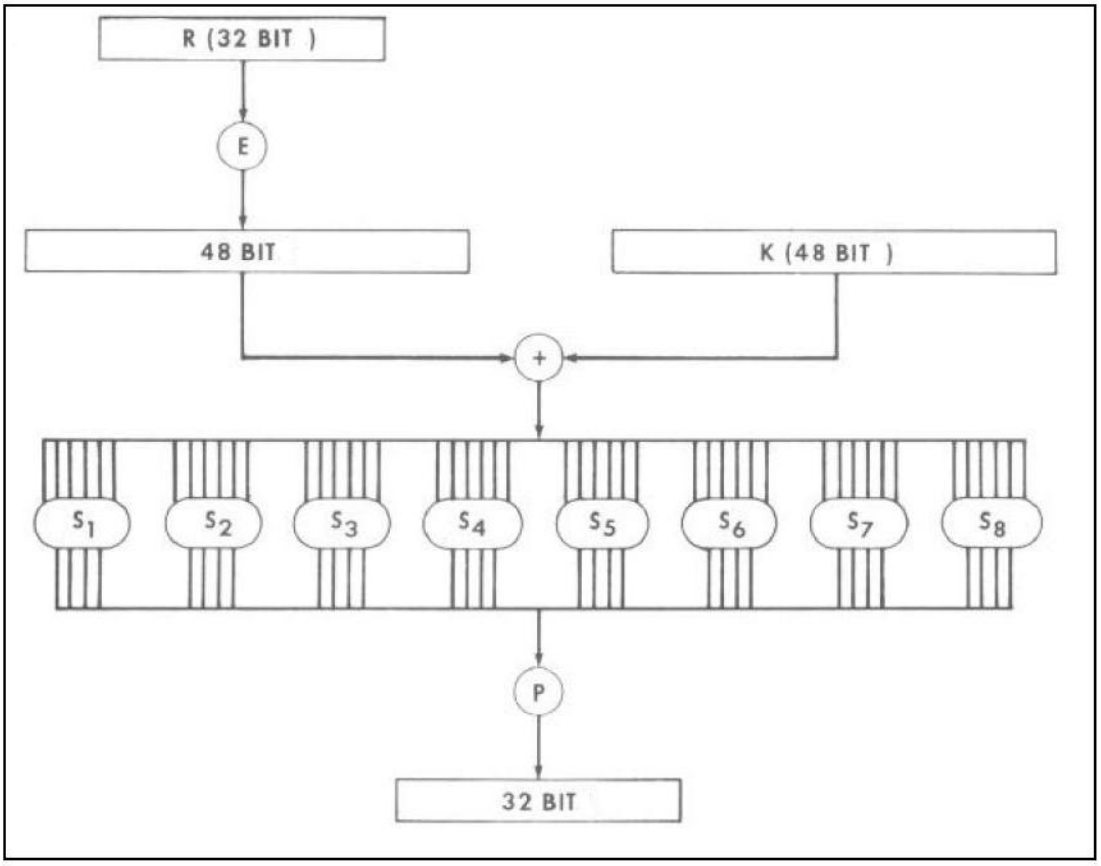
$$L' = R \tag{3.3}$$

$$R' = L \oplus f(R,K) \tag{3.4}$$

Her DES turunda bulunan DES *f* fonksiyonu şekil 3.4'te verilmiştir.



Şekil 3.3 : DES Kutusu ve Turların Dağılımı.



Şekil 3.4 : DES *f* fonksiyonu.

Şekilden de görüldüğü gibi, var olan turdaki 32 bitlik sağ yarı blok E (Expansion) fonksiyonu ile 48 bite genişletilir. DES *f* fonksiyonu içindeki E fonksiyonu şöyledir [4]:

E FONKSİYONU

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

E fonksiyonu çıkışındaki 48 bit, 48 bitlik anahtar ile e-xor' lanarak S-Kutusu girişlerini oluşturur. S-Kutularına 6 bit girer ve 4 bit çıkar. S-Kutularının hangi girişe

karşılılık hangi çıkışı vereceği [4]' te bulunabilir. 6 bitlik girişin ilk ve son bitleri s-kutusundaki çıkışın satır numarasını, girişin ortada kalan 4 biti ise çıkışın sütun numarasını verir. S-kutusunda karşılık gelen değer 4 bitlik çıkışı verir.

8 adet 4 bitlik s-kutusu çıkışları son olarak olarak permütasyondan geçer. DES *f* fonksiyonu içindeki permütasyon şu şekilde bit yer değişimi yapar [4]:

P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Son permütasyon çıkışındaki 32 bit, DES *f* fonksiyonunun çıkışı olur.

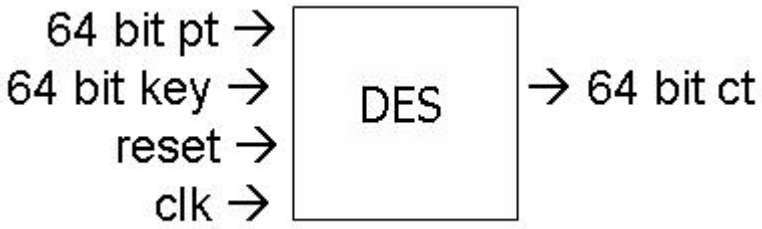
Böylelikle DES algoritmasındaki bütün fonksiyonlar tamamlanarak 64 bitlik anahtar ile 64 bitlik düz metin, 64 bitlik şifreli metne dönüşmüş olur.

3.3 Gerçekleme Adımları

DES Blok Şifreleme Algoritmasının FPGA üzerinde düşük enerjili tasarımı yüksek lisans tez çalışmasının ilk aşaması olarak VHDL donanım betimleme dilinin öğrenilmesi gelmektedir. Bilinen en güncel donanım betimleme dillerinden biri olan VHDL' nin öğrenilmesinde [13]' ten yararlanılmıştır.

Gerçekleme aşamasında VHDL dili kullanılmış olup, FPGA'da sentezleme, yerleştirme ve yollandırma aşamaları için Xilinx ISE 9.2i programı kullanılmıştır. Ayrıca simülasyon aşamasında Modelsim 6.3c ve güç ölçümleri için de XPOWER aracı kullanılmıştır.

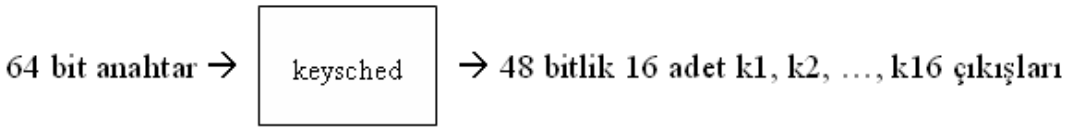
DES algoritmasının VHDL ile yazılımı için öncelikle [4] dökümanı çok iyi anlaşılmış ve daha sonra algoritmanın yazılmasına başlanmıştır. Bölüm 3.2' de anlatılan yapıların VHDL kodu yazılmıştır. Kodun tamamının bir dosyada yazılması yerine içiçe fonksiyonlar yazılarak kodun anlaşılması ve yazılması kolaylaştırılmıştır. DES ana fonksiyonunun blok şeması şekil 3.5' te verilmiştir. DES kutusuna 64 bitlik düz metin ve anahtarla birlikte saat ve yeniden başlatma işaretleri girer, 64 bitlik şifreli metin çıkar.



Şekil 3.5 : DES ana fonksiyonu.

DES algoritmasında kullanılan temel alt bloklar şunlardır: keysched (anahtar üretim) bloğu, ip (ilk permütasyon) bloğu, roundfunc (tur fonksiyonu) bloğu ve fp (ters ilk permütasyon) bloğudur.

keysched alt bileşenine 64 bitlik anahtar girer ve her bir turda kullanılmak üzere 48 bitlik 16 adet anahtar çıkar. Şekil 3.6' da blok şema verilmiştir. keysched bileşeni iki alt bileşenden oluşur: PC1 ve PC2.

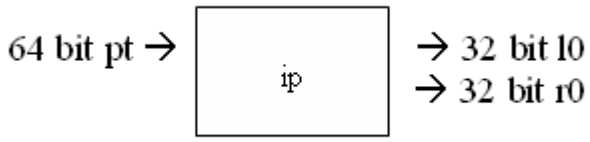


Şekil 3.6 : Anahtar üretim fonksiyonu.

PC1 alt bileşenine 64 bitlik anahtar girer ve 56 bit olarak çıkar. PC1 alt bileşeninde permütasyon ve sola kaydırma işlemleri yapılır. Atılan 8 bit hata veya benzerlik tanımlama için kullanılır.

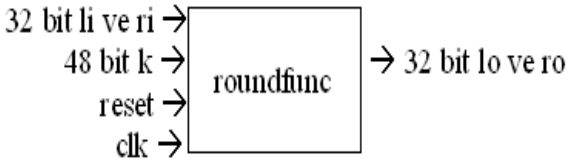
PC2 alt bileşenine ise 56 bit girer ve 48 bitlik anahtar çıkar. Bu alt bileşende de yine permütasyon yapılır.

ip alt bileşeninde permütasyon yapılır. 64 bitlik düz metin girer, 32 bitlik sol ve sağ yarı blokları çıkar. Şekil 3.7' de blok şeması verilmiştir.



Şekil 3.7 : İlk permütasyon fonksiyonu.

Üçüncü temel alt bileşen olan tur fonksiyonu, DES kutusunun en temel bileşenidir. Şekil 3.8' de blok şeması verilmiştir. Tur fonksiyonu bileşenine 32 bitlik sol yarı ve sağ yarı blokları ile birlikte 48 bitlik anahtar, saat ve yeniden başlatma işaretleri girer, 32 bitlik sol yarı ve sağ yarı blokları çıkar.



Şekil 3.8 : DES tur fonksiyonu.

DES tur fonksiyonu bileşeninin 13 adet alt bileşeni vardır.

xp (genişletme) alt bileşeninde 32 bitlik giriş, bazı bitler tekrarlanarak ve permütasyon yapılarak 48 bitlik çıkış elde edilir.

desxor1 alt bileşeninde xp'den çıkan 48 bit ile 48 bitlik anahtar e-xor'lanır. Sonuçta 8 adet 6'şar bitlik çıkış s-kutularına giriş olur.

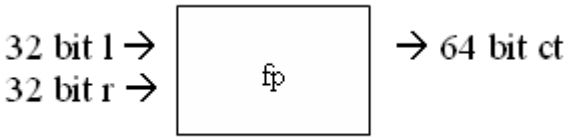
s1,s2,…,s8 s-kutularına 6 bit girip 4 bit çıkar. Bir çeşit LUT görevi yaparlar.

pp (son permütasyon) alt bileşeninde yine permütasyon yapılır. 8 adet dörder bitlik s-kutusu çıkışı girer ve 32 bit olarak permütasyon sonucu çıkar.

desxor2 alt bileşeninde ise yine pp çıkışı ile bir önceki turun sol yarısı e-xor'lanır.

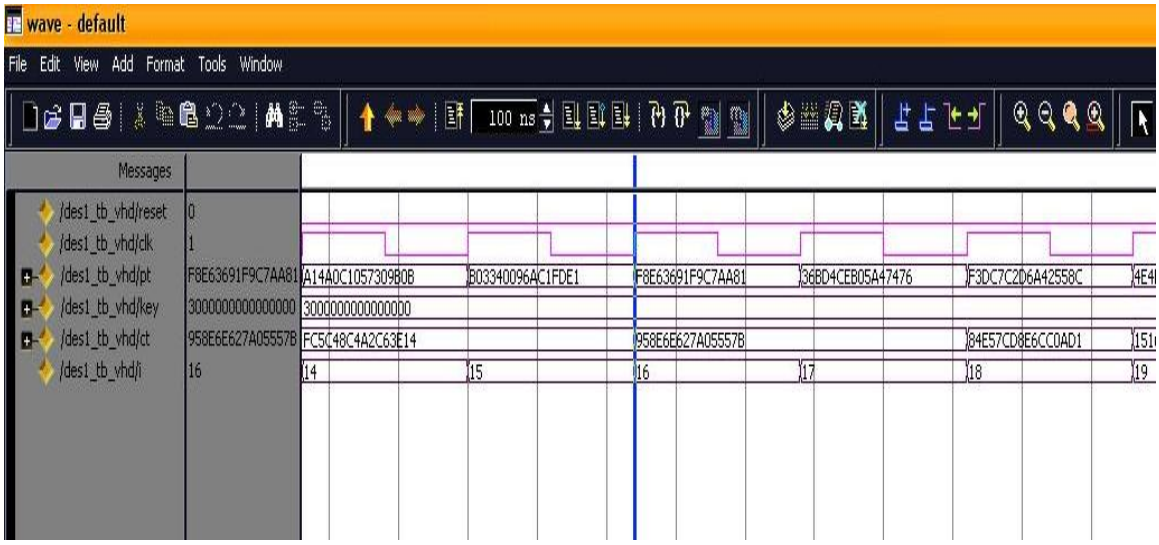
reg32 alt bileşeninde ise her bir turda kullanılan kaydedici yapısı vardır.

Son alt bileşen olan fp (ters ilk permütasyon) bileşeninde ise adından da anlaşılacağı üzere ip' de yapılan permütasyonun tersi yapılır. 32 bitlik sol ve sağ yarı blokları girer, 64 bitlik şifreli metin çıkar. Şekil 3.9' da blok şeması verilmiştir.



Şekil 3.9 : Ters ilk permütasyon fonksiyonu.

DES algoritmasının kodu yazıldıktan sonra, şekil 2.1' deki tasarım sürecine uygun olarak fonksiyonel benzetim (simülasyon) yapılmıştır. ModelSim programı yardımıyla yapılan simülasyon sonucu kodun önceden bilinen üç adet giriş çıkış değeri için doğru çalıştığı gözlenmiştir. Şekil 3.10' da ModelSim programında elde edilen simülasyon sonucu verilmiştir.



Şekil 3.10 : DES simülasyon sonucu.

Yazılan kodun doğru çalıştığına emin olunduktan sonra, tasarım sürecinde bir sonraki adım olan FPGA üzerinde gerçeklemeye geçilmiştir.

3.4 FPGA Üzerinde Gerçekleme

DES Blok Şifreleme Algoritmasının FPGA üzerinde düşük enerjili tasarımı yüksek lisans tez çalışmasının önemli bir aşaması da yazılan VHDL kodlarının FPGA donanımı üzerinde gerçeklenmesidir. Gerçeklemeler sırasında devrelerin alan, zaman, iş hacmi (throughput), güç ve enerji bilgileri toplanmıştır. Bu çalışmada bölüm 2.2' de de bahsedildiği gibi Xilinx firmasına ait Spartan-3 XC3S5000 FPGA' sı kullanılmıştır. Çalışma sırasında farklı DES yapıları gerçeklenmiş ve karşılaştırılmıştır. Adil bir karşılaştırma olması amacıyla bölüm 5.2' de anlatılacak olan bütün gerçeklemelerde aynı adımlar takip edilmiştir.

Yazılan bir VHDL kodunun doğruluğu yapılan simülasyon sonucu anlaşıldıktan sonra FPGA üzerinde gerçekleme işlemi Xilinx ISE 9.2i programında kodun sentezlenmesiyle başlar. Sentezleme aşamasında dikkat edilmesi gereken noktalar vardır. Yazılan kodun çalışma amacına bağlı olarak, sentezleme seçeneklerinden optimizasyon hedefi (optimization goal) olarak hız veya alan seçilir. Ayrıca optimizasyon girişimi (optimization effort) nomalden yükseğe çekilerek FPGA üzerinde en iyi gerçekleme sağlanmıştır. Sentezleme sonucunda devre ile ilgili zaman ve alan bilgileri elde edilir. Fakat bu değerler sentezleme sırasındaki tahmini değerlerdir, gerçek değerler için yerleştirme ve yollandırma (place & route) sonrasındaki alan ve zaman bilgilerine bakmak gerekir. Örneğin işhattı (pipeline) gerçekleme için sentezleme sonucu alan ve zaman bilgileri şöyledir:

Device utilization summary:

Selected Device : 3s5000fg900-5

Number of Slices:	**2129 out of 33280 6%**
Number of Slice Flip Flops:	**1024 out of 66560 1%**
Number of 4 input LUTs:	**4123 out of 66560 6%**
Number of IOs:	194
Number of bonded IOBs:	186 out of 633 29%
Number of GCLKs:	1 out of 8 12%

Timing Summary:

Speed Grade: -5

Minimum period: 5.628ns (Maximum Frequency: 177.688MHz)
Minimum input arrival time before clock: 6.936ns
Maximum output required time after clock: 6.216ns
Maximum combinational path delay: No path found

Görüldüğü gibi devrenin çalışabileceği en yüksek saat frekansı yaklaşık 178 MHz (Mega-Hertz) çıkmıştır. Halbuki gerçek değerler yerleştirme ve yollandırma sonucu elde edilecektir.

Sentezleme işleminden sonra FPGA üzerine dönüştürme (translate), eşleştirme (map), yerleştirme ve yollandırma (place & route) işlemleri gerçekleştirilmiştir. Ayrıca diğer bir önemli nokta da kullanıcı kısıtlamalarıdır (user constraints). Kullanıcı kısıtlamaları olarak saat frekansı verilmiş ve devrenin çalışabileceği en yüksek saat frekansı iterasyonla bulunmuştur. Bu işlemlerin sonucunda oluşan zamanlama ve eşleştirme raporlarından devrenin gerçek alan ve zaman bilgileri elde edilmiştir. Yine örnek olarak işhattı gerçeklemesi için alan ve zaman bilgileri şöyledir:

Design Summary

Logic Utilization:
Number of Slice **Flip Flops:** **928** out of 66,560 1%
Number of 4 input **LUTs:** **4,125** out of 66,560 6%
Logic Distribution:
Number of occupied **Slices:** **2,320** out of 33,280 6%

Timing summary:

Design statistics:
Minimum period: 9.796ns (Maximum frequency: 102.082MHz)

Görüldüğü gibi alan ve zaman bilgileri sentezleme sonucuna göre farklılık göstermektedir.

Devrelerin harcadığı güç ve enerji bilgilerini elde edebilmek için yerleştirme ve yollandırma sonrası simülasyon (post-route simulation) yapılır. Bu simülasyon sırasında devreye MATLAB programı tarafından oluşturulmuş rastgele girişler uygulanır ve devre mümkün olan en yüksek saat frekansında çalıştırılarak simülasyon sonuçları elde edilir. Burada dikkat edilmesi gereken nokta, yerleştirme ve yollandırma sonrası simülasyonda kullanılacak saat frekansıdır. Yerleştirme ve yollandırma sonucu elde edilen rapordaki en yüksek saat frekansında devre içindeki kapı gecikmelerine bağlı olarak devre doğru çıkışları üretemeyebilir. Çıkışların doğruluğu kontrol edilerek, mümkün olan en yüksek saat frekansı iterasyonla bulunmalıdır. Örneğin işhattı gerçeklemesi için saat frekansı 102 MHz verilerek yapılan yerleştirme ve yollandırma sonrası simülasyon sonucu çıkışlar yanlış

olmaktadır. Bu sebeple saat frekansı iterasyonla düşürülerek 83 MHz verilmiş ve yapılan simülasyon doğru sonuçları vermiştir. Simülasyon sırasında güç analizi işleminde kullanılmak üzere .vcd uzantılı simülasyon dosyası da oluşturulmuştur. Güç analizi adımları için [3] no'lu kaynaktan yararlanılmıştır. Son olarak elde edilen simülasyon dosyası kullanılarak XPOWER güç ölçüm aracı yardımıyla devrenin harcadığı güç bilgisi elde edilmiştir. Örneğin işhattı gerçeklemesi için güç değeri 48,7 mW (mili-watt) çıkmıştır. Böylelikle yazılan VHDL kodunun FPGA üzerinde gerçeklenmesi aşaması tamamlanmıştır. Bununla birlikte farklı DES yapıları için elde edilen alan, zaman, iş hacmi (throughput), güç ve enerji bilgileri bölüm 5.2 ve 5.3' te detaylı olarak anlatılacaktır.

4. DÜŞÜK GÜÇ YÖNTEMLERİ

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea

Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

Şekil 4.1 : Örnek şekil.

This indicates that the ANN is accurate at base flow and flow height values lower then 3 m.

Çizelge 4.1 : Çizelge örneği.

Kolon A	Kolon B	Kolon C	Kolon D
Satır A	Satır A	Satır A	Satır A
Satır B	Satır B	Satır B	Satır B
Satır C	Satır C	Satır C	Satır C

Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

5. DÜŞÜK ENERJİLİ DES GERÇEKLEMESİ

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

5.1 Önceki Çalışmalar

In this thesis, the necessary steps for constructing an end-to-end streamflow forecasting system were discussed. These steps include the use

5.2 Farklı DES Yapıları

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea

5.2.1 İşhattı (Pipeline) Gerçeklemesi

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea

5.2.2 Kaydedicisiz Gerçekleme

Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

5.2.3 Klasik Yapı Gerçeklemesi

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Şekil 5.1 : Beşinci bölümde bir örnek şekil.

This indicates that the ANN is accurate at base flow and flow height values lower then 3 m.

Çizelge 5.1 : Beşinci bölümde bir örnek çizelge.

Kolon A	Kolon B	Kolon C	Kolon D
Satır A	Satır A	Satır A	Satır A
Satır B	Satır B	Satır B	Satır B
Satır C	Satır C	Satır C	Satır C

5.2.4 Sekiz Turda İki S – Kutusu ile Gerçekleme

Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

5.2.5 Sekiz Turda Tek S – Kutusu ile Gerçekleme

In this thesis, the necessary steps for constructing an end-to-end streamflow forecasting system were discussed. These steps include the use

5.2.6 İç ve Dış Kaydedicili Yol Yapısında Gerçekleme

Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

5.3 Farklı DES Yapılarının Karşılaştırılması

Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

6. SONUÇLAR VE TARTIŞMA

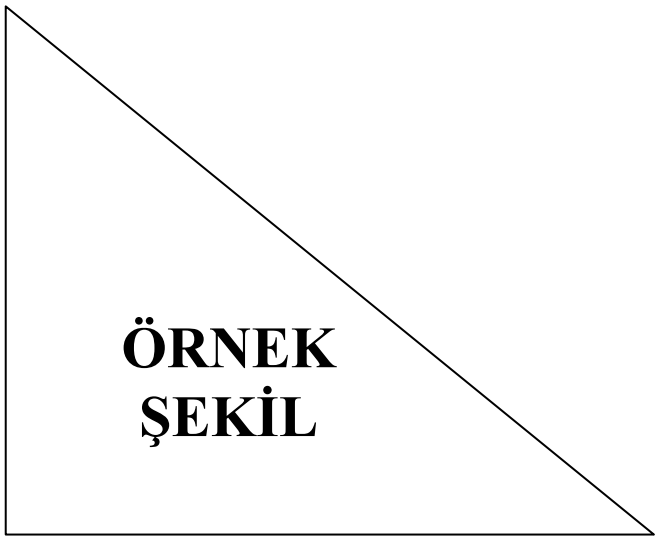
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gub rgren, no sea

Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.



Şekil 6.1 : Altıncı bölümde bir örnek şekil.

This indicates that the ANN is accurate at base flow and flow height values lower then 3 m.

Çizelge 6.1 : Altıncı bölümde bir çizelge.

Kolon A	Kolon B	Kolon C	Kolon D
Satır A	Satır A	Satır A	Satır A
Satır B	Satır B	Satır B	Satır B
Satır C	Satır C	Satır C	Satır C

Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna. Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

Stet clita kasd gub rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut lab ore sit et dolore magna.

KAYNAKLAR

[1] **Savaş, E.,** 1994. Dizi Şifreleme Sistemleri ve Doğrusal Karmaşıklık, *Yüksek Lisans Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[2] **Acar, S.,** 2005. Eliptik Eğri Kriptografisinde Skaler Çarpma Bloğunun VHDL ile Tasarımı, *Yüksek Lisans Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[3] **Doğan, A.Y.,** 2008. AES Algoritmasının FPGA Üzerinde Düşük Güçlü Tasarımı, *Yüksek Lisans Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[4] **FIPS 46-3,** 1999. Data Encryption Standard. National Institute of Standarts and Technology (NIST).

[5] **Lagger, A.,** 2002. Implementation of DES Algorithm Using FPGA Technology, *Semester Project*, EPFL, Lozan.

[6] **Berna, A.,** 1998. Sahada Programlanabilir Kapı Dizileri ile Lojik Devre Tasarımı ve VHDL Kullanılarak Bazı Devrelerin Gerçekleştirilmesi, *Yüksek Lisans Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[7] **Topçu, İ.H.,** 2002. Sahada Programlanabilir Kapı Dizileri Kullanılarak Sayısal Tasarım Kartı Gerçeklenmesi, *Yüksek Lisans Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[8] **Ordu, L,** 2006. AES Algoritmasının FPGA Uzerinde Gerceklenmesi ve Yan Kanal Analizi Saldırılarına Karsı Guclendirilmesi, *Yüksek Lisans Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[9] **Xilinx**. Spartan – 3 Field Programmable Gate Arrey Family Data Sheet. http://www.xilinx.com/support/documentation/data_sheets/spartan3

[10] **Kaplan, T.,** 2006. Trivium Dizi Şifreleme Algoritmasının FPGA Üzerinde Gerçeklenmesi, *Lisans Bitirme Ödevi,* İ.T.Ü. Elektrik-Elektronik Fakültesi, İstanbul.

[11] **Şahinoğlu, M.,** 2008. AES Algoritmasının FPGA Üzerinde Gerçeklemesine Elektromanyetik Alan Saldırısı, *Yüksek Lisans Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

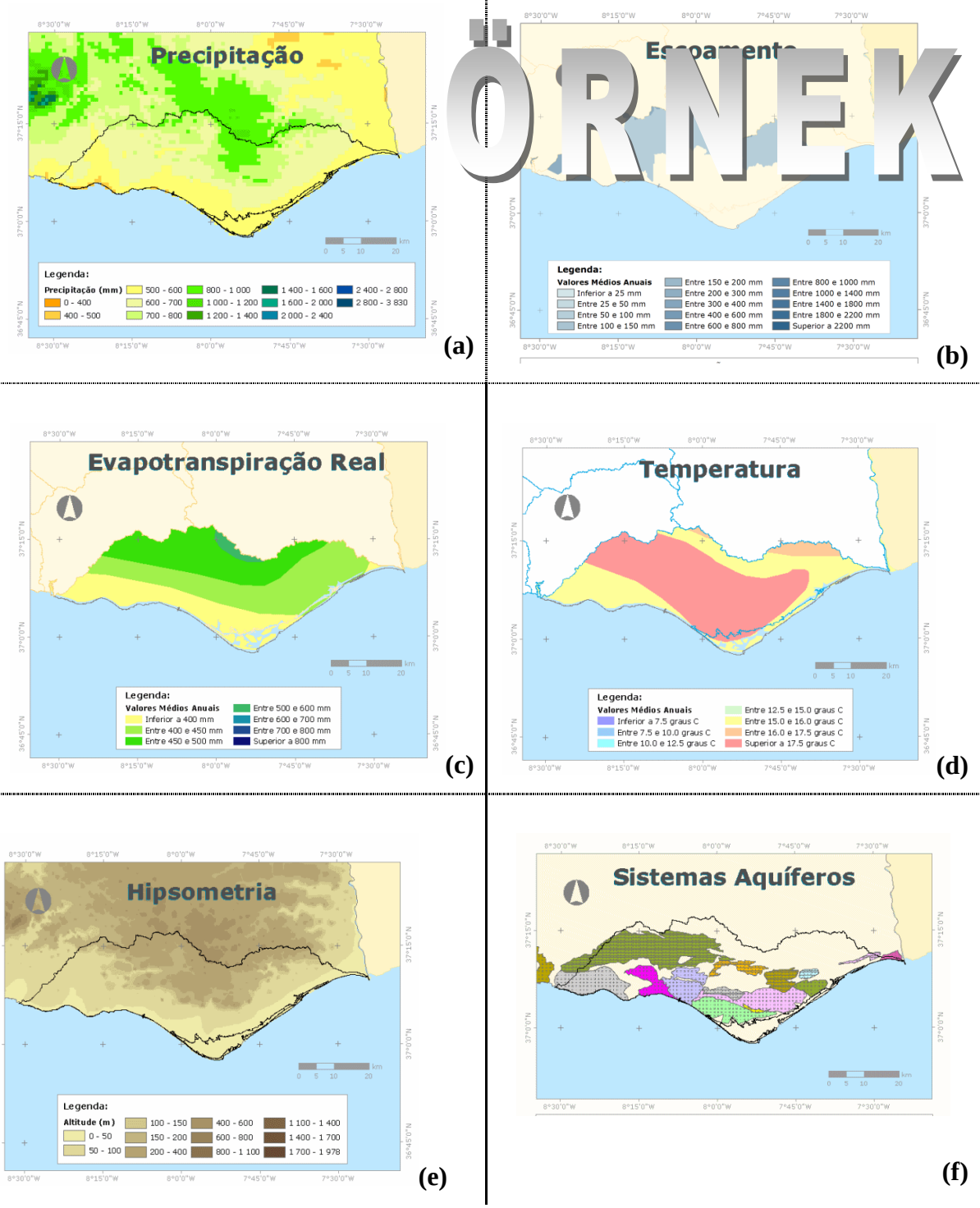
[12] **Stinson, D.R.,** 2002. Cryptography, Chapman & Hall/CRC Press Company, sf. 95-102, United States of America.

[13] **Hsu, Y.C., Tsai, K.F., Liu J.T. and Lin, E.S.,** 1995. VHDL Modeling For Digital Synthesis, Kluwer Academic Publishers, Riverside.

EKLER

EK A.1 : Haritalar

EK A.1



Şekil A.1 : Bölgesel haritalar: (a)Yağış. (b)Akım. (c)Evapotranspirasyon …

Çoklu şekillerde herbir farklı şekil, gerekiyorsa (metin içinde herbirine birine atıf yapılacaksa) teker teker harflendirilerek ve açıklamasıyla verilir. Genel bir isim yeterli ise harflendirmeye **eklerde** gerek duyulmaz.
Bu bir nottur. cıktı alamadan önce siliniz.

Çizelge A.1 : Ekler bölümünde çizelge örneği.

Kolon A	Kolon B	Kolon C	Kolon D
Satır A	Satır A	Satır A	Satır A
Satır B	Satır B	Satır B	Satır B
Satır C	Satır C	Satır C	Satır C

ÖZGEÇMİŞ

Ad Soyad: Tarik KAPLAN

Doğum Yeri ve Tarihi: Antalya - 1983

Lisans Üniversite: İstanbul Teknik Üniversitesi, Elektronik Mühendisliği, 2006